MODELING LIGHT PENETRATION THROUGH A SLASH PINE (*Pinus elliottii*)CANOPY

By

KEVIN SCOT MCKELVEY

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1990

UNIVERSITY OF FLORIDA LIBRARIES

This work is dedicated to my wife Madeleine, and my two daughters Adrienne and Tara.

ACKNOWLEDGEMENTS

Much of the data in this dissertation was the result of the work of many: Collection of data on CO_2 assimilation, foliar biomass, and light absorption by the canopy were carried out as part of a four-year study of slash pine funded by the National Science Foundation. In particular I would like to thank Henry Gholz and Sherry Vogel for the analysis of biomass fluctuation, Wendell Cropper for the analysis of assimilation data, and Katherine Ewel for the long hours of consultation without which this work would not have progressed. If there is good science here, she should receive a great deal of the credit.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii

ABSTRACT v

INTRODUCTION 1

MODELING HIERARCHICAL EFFECTS IN SLASH PINE 14
Characteristics of Slash Pine 14
Evaluation of Model Approaches 17
Light Penetration Modeling in Discrete Space 23

A DISCRETE-SPACE MODEL FOR SLASH PINE 29
Data Acquisition 29
Model Structure 45
Model Results 53

MODELING SEASONAL BELOW-CANOPY LIGHT PENETRATION 62

MODELING LIGHT PENETRATION WITHIN THE CANOPY 88

EVALUATING THE MODEL 95

CONCLUSIONS 101
Discussion of Light Penetration
Through Slash Pine 101
Future Research Needs 106

LITERATURE CITED 107

APPENDIX COMPUTER CODE FOR DISCRETE-SPACE
MODELING OF LIGHT PENETRATION 110

BIOGRAPHICAL SKETCH 136

Abstract of Dissertation Presented to the Graduate School of the University of Florida in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

MODELING LIGHT PENETRATION THROUGH A SLASH PINE (Pinus elliottii) CANOPY

By

Kevin Scot McKelvey

August, 1990

Chairman: Katherine Ewel
Major Department: School of Forest Resources and Conservation

Conifer canopies are a hierarchy of discrete objects: crowns, shoots, and needles. Light penetration through this hierarchy is controlled by the size, arrangement, and opacity of objects at each level. If light penetration patterns are dominated by organization at the shoot or the crown level, the relationship between light penetration and leaf area will be indirect. The crowns of slash pine are conic, and the foliage is clumped. A discrete shoot-level geometric model was created to determine the importance of this structure on light penetration.

Shoot-level parameters were measured in the laboratory using controlled light. Light penetration data was gathered from sensor arrays placed above and below the canopy. Seasonal foliage biomass patterns for the stand were derived from destructive sampling, litterfall, and needle elongation rates.

Light penetration through the canopy was simulated for the full range of sun angles and azimuths. Azimuth was not found to be an important controlling parameter for light penetration, and model output reduced to a single multiple regression equation based on sun angle and biomass per shoot. This simplified model was used to simulate light penetration for 505 days.

Data indicated that light emanated primarily from the mid-sky on uniformly overcast days. Diffuse light was simulated by assuming that all diffuse light emanated from 45-degree angles of elevation.

The model duplicated observed patterns of seasonal variation in below-canopy light penetration and explained 83% of the daily variation in below-canopy light penetration during a 2-month period centered on the winter solstice.

Light penetration within the canopy, when simulated for planar surfaces, was similar to patterns generated by vertically stratified, horizontally homogeneous models. When light penetration to the simulated shoot positions was modeled, however, the patterns diverged due to self-shade within the crown.

The model accurately tracked the pattern of light penetration over time and, once calibrated, could be used to predict seasonal changes in foliar biomass. Leaf area could not, however, be accurately estimated directly from light

penetration because the proportion of light interception due to woody biomass was unknown.

INTRODUCTION

Photosynthesis is the only pathway by which sun energy can enter an ecosystem and be stored in a metabolically utilizable form. Ecosystem metabolism is dependent upon and limited by photosynthetic production. Obtaining an accurate estimate of photosynthetic input is therefore essential to any attempt to quantify energy or carbon flow through an ecosystem.

The actual measurement of gross primary productivity (GPP) is difficult due to photorespiration. For most practical purposes, however, the rate of CO_2 assimilation, which is measurable, provides a good approximation of carbon input into the system. The intensity of photosynthetically active radiation (PAR) at the leaf surface acts as a direct control on assimilation rates. Because canopy-level assimilation is the sum of the assimilation rates of all of the leaves, leaf assimilation can be scaled up to the canopy level if the quantity of foliage and the light penetration patterns through the foliage are known.

Over time, foliage position and light penetration are coupled. A change in foliage distribution will cause a shift in the light penetration pattern. A shift in the light penetration pattern will in turn cause a change in foliage distribution due to growth and abscission. Accurate light penetration formulations are therefore a necessary component of either static or dynamic physiological models.

Life is defined by pattern, and pattern exists at all levels of scale in nature. In science in general and modeling in particular, pattern is often ignored in favor of mean values. Models based on mean values are simpler and less data intensive, but there are dangers in this approach if relationships between variables are nonlinear.

In light penetration modeling through forest canopies, foliage is generally assumed to lack spatial pattern, and mean values for light penetration through the canopy are used indirectly to derive a variety of parameters such as leaf area index (LAI), photosynthesis, and evapotranspiration. Because all of these parameters are related to light penetration through nonlinear functions, heterogeneous foliage distribution can affect the calculated value of these parameters. This dissertation examines the effect of foliage distribution patterns on light penetration through a 20-year-old slash pine (Pinus elliottii) plantation in north Florida.

Traditional light penetration models assume that foliage distribution is homogeneous in space and that leaves are opaque. If this assumption of opacity is valid, the probability of a beam of light penetrating the canopy is simply the probability that it will pass through the canopy without contacting any of the leaves. If the leaves are randomly distributed, the number of contacts follows the Poisson distribution with

$$P_{\text{no contacts}} = \exp(-uz) \qquad (1)$$

where

$P_{\text{no contacts}}$ = the probability that a beam passes through the canopy without intersecting any leaves,

u = the average leaf area, and

z = the length of the path that the beam must travel (Oker-Blom 1986).

For a large number of beams, $P_{\text{no contacts}}$ becomes the average proportion of light beams penetrating to depth z through the canopy. This equation, generally known as Beer's law, is often expressed as

$$I_z = I_o \exp(-kuz) \qquad (2)$$

where

I_z = the photon density at depth z,

I_o = the photon density above the canopy, and

k = the ratio projected to measured leaf area.

Oker-Blom (1983) argues that, in conifers, shoots should be considered to be discrete entities. If the shoots are randomly distributed and opaque, then equation 2 is valid for shoots as well as for needles. In this case, light penetration through shoots will differ from light penetration through needles to the extent that the density of the shoots differs from the density of the needles (Carter and Smith 1985).

This concept can be expanded to include all potential distribution patterns and scales of patterning. Conifer canopies consist of a hierarchy of discrete objects distributed in space, from individual crowns down to opaque particles within the needles. At each level, the distribution of foliage is discontinuous. All the foliage in the canopy is contained within individual crowns which are generally conic or paraboloid in form. Within the crowns, foliage is confined to the regions adjacent to actively growing meristems (shoots), and within these shoots, foliage is confined to volumes defined by individual needles. These levels of hierarchy are discrete in space because each divides space into distinct volumes

in which foliage can exist and where the probability of existence is zero. For crowns the foliage is confined to a volume defined by branch length and inclination. Moving down through the hierarchy, each level is spatially bounded by all of the higher levels. Shoot position is confined within the crown, needle position within the shoot.

For any discrete array of n objects in space, the proportion of light penetrating through the array is

$$K_{array} = P_0 + P_1K_1 + P_2K_2 + \ldots P_nK_n \qquad (3)$$

where

K_{array} = the total light penetrating the array,

P_0 = the probability that no object is encountered,

$P_1 \ldots P_n$ = the probabilities that 1...n objects are encountered, and

$K_1 \ldots K_n$ = the proportion of light penetrating the objects given that 1...n objects are encountered.

If it is assumed that all objects are identical, then K_{object} is defined as the average proportion of light penetrating through an individual object. Equation 3 then simplifies to

$$K_{array} = P_0 + \sum_{i=1}^{n} P_i K^i_{object} \qquad (4)$$

If K_{object} is small (the objects are opaque), then K_{array} is approximately equal to P_0. P_0 is entirely dependent on the distribution and density of the objects and not on their internal structure. If P_0 dominates, then light penetration through the canopy will be controlled by the distribution pattern of objects in the canopy.

Because conifer canopies are composed of a hierarchy of discrete objects, equation 4 can be applied repeatedly at each level to define overall light penetration through the canopy:

$$K_{canopy} = P_0 + \sum_{i=1}^{n} P_{i,crown} K^i_{crown} \qquad (5)$$

$$\downarrow$$

$$K_{crown} = P_0 + \sum_{i=1}^{n} P_{i,shoot} K^i_{shoot}$$

$$\downarrow$$

$$K_{shoot} = P_0 + \sum_{i=1}^{n} P_{i,leaf} K^i_{leaf}$$

$$\downarrow$$

$$K_{leaf} = P_0 + \sum_{i=1}^{n} P_{i,cell} K^i_{cell}$$

When the canopy is viewed in terms of these equations it is possible to define the precise effect of each level of hierarchy on the overall light penetration through the

canopy and the potential errors involved in assuming homogeneity at any level.

Any level of the hierarchy is homogeneous if the density of the objects that define the level is evenly distributed. If the average density, u_{object}, is of integer order (1,2,3, etc.), then homogeneity can be defined as a special case of equation 4: all light beams encounter an equal number of objects. As a system approaches homogeneity, all the terms in the series except the term defining the mean number of encounters vanish, P_{mean} approaches 1.0, and the system collapses to the next lower level in the hierarchy.

If K_{object} is approximately equal to zero at any level, then the effects of all lower levels will vanish. This has traditionally been the assumption for light penetration through leaves. Equation 2, therefore, simply represents a special case: $K_{leaf} = 0$, leaves are randomly distributed, and homogeneity is assumed for all higher levels (shoot and crown) of the hierarchy.

If patterns exist above the leaf level and are ignored, then light penetration through the canopy will always be underestimated. The extent of underestimation will primarily be a function of the magnitude of the P_0 terms because, with the assumption of homogeneity, these terms are assumed to be zero. Because the P_0 terms can easily

dominate their respective series, errors of omission can be large. If, for instance, a stand has a measured u = 4.0, k = 0.5, z = 1.0, with random distribution at the leaf level and homogeneity at the shoot and crown level, the calculated light penetration (equation 2) for vertical light is $0.135*I_o$. If, however, the foliage is held on randomly distributed shoots with an internal leaf area of 4.0 and a projected shoot area of 1.0, then, from equation 3, the anticipated light penetration is

$$K_{shoot} \approx 0.37 + 0.37*0.135 + 0.18*0.0182 + \ldots \approx 0.42 \quad (6)$$

In this case, three times as much light is expected to penetrate through random shoots even though the mean projected leaf area is constant in both cases. This is largely due to the magnitude of the P_o term (0.37). P_1, equal to 1.0 in the homogeneous case, is only 0.37 in the canopy composed of randomly placed shoots.

In a field of crowns, crown structure and position will dominate light penetration if the crowns are widely spaced, and $P_{no\ crown}$ will approach 1.0 as the ratio of space between crowns to crown diameter becomes large. For this reason, discrete crown models have been used to simulate row crops and orchards (Mann et al., 1980; Norman and Welles, 1983; Rook et al., 1985; Whitfield, 1986; Cohen and Fuchs, 1987). For a field of conic crowns, this ratio is minimized at the

point of crown closure and increases with height. $P_{no\ crown}$ therefore shifts with height. For vertical light entering an idealized closed stand composed of uniform conic crowns, $P_{no\ crown}$ equals $\pi/4$ at the point of crown closure and converges rapidly on 1.0 with increasing height (Figure 1). In species with conic or parabolic crowns, light penetration patterns in the upper crown may therefore be dominated by crown form even if homogeneity is a reasonable assumption for light penetration to the forest floor.

Leaf area is an important parameter for determining the magnitude of canopy level physiological processes, but it is difficult to measure directly. It is common practice to sample light above and below the canopy and, assuming equation 2 to be valid, to compute leaf area (Lang and Yuequin, 1986; Pierce and Running, 1988; Campbell and Norman 1989) by inverting the function. The problem with this approach is that, if there is canopy structure at the shoot and crown levels, then k loses its definition as the ratio of projected to measured leaf area.

For leaves, k is generally close to 0.5, the theoretical value if the angular distribution of leaves is random (Campbell and Norman 1989). This figure, or one very close, is often used in equation 2 to determine leaf area index (LAI). In canopies with highly opaque clusters, however, the value of k will become a function of the

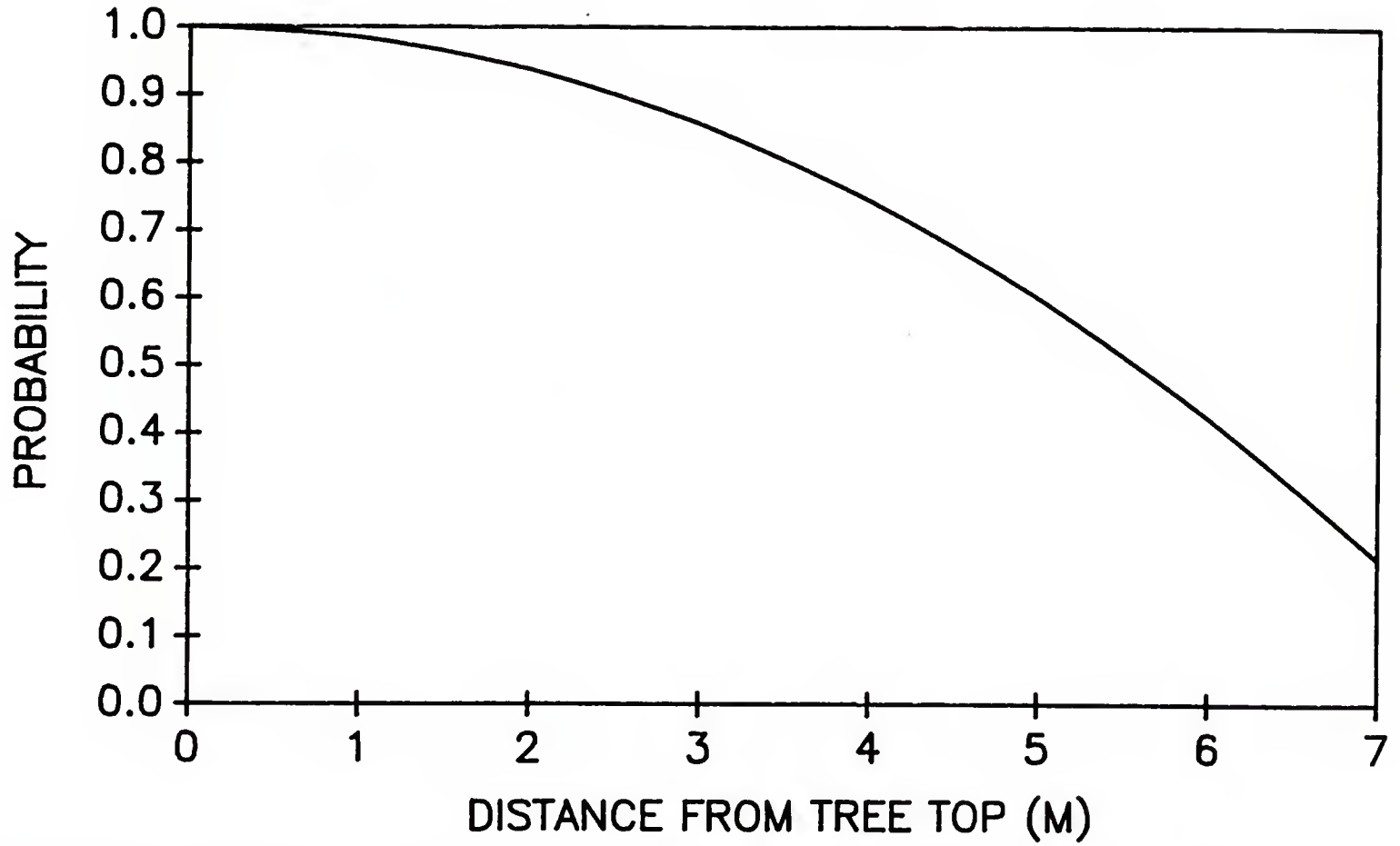

Figure 1: The probability that vertical light will not intersect a crown as a function of height. An idealized conic crown form is assumed, with 3x3 m spacing and a crown height of 7 m.

distribution and shape of the clusters and will be only indirectly related to leaf area.

In general, the effects of various scales of patterning on light penetration are difficult to separate at the forest floor, and k may have no clear geometrical interpretation. This does not necessarily mean that Beer's law doesn't 'work' for the stand, simply that there may be uncertainty as to which level of hierarchy is controlling its parameters. Mistakes in understanding these hierarchical effects can lead to a variety of errors when using inversion techniques. If, for instance, in the example listed above (equation 6), the leaf area was unknown and the k was assumed to be 0.5, the calculated leaf area for the stand based on light penetration to the forest floor would be

$$-\ln(0.42)/0.5 = 1.74 \qquad (7)$$

rather than the actual value of 4.0.

Predictions based on inversion are prone to similar errors. The effects of new foliage growth on light penetration, for instance, depend on the impact of new leaf production on the P_0 terms at each level in the hierarchy. In a clustered system, if new foliage enters the system at locations spatially independent of the existing foliage and

if it enters in the form of new shoots of constant dimension, then shoot area will be related to leaf area through a constant. If cluster location is random, a slightly modified Beer's law will approximate light penetration for all leaf densities:

$$I_z = I_o \exp(-kuzq) \qquad (8)$$

where

q = a clustering constant with the geometrical interpretation of the ratio of measured leaf area to projected cluster area (Campbell and Norman 1989).

Spatial independence of new foliage in conifers is, however, biologically improbable. Because both branch and foliage production occur exclusively at the branch tips, new foliage will enter the system at spatial locations determined largely by the position of the existing foliage. This means that the position of the new foliage in space will be conditioned by the position of the existing foliage, and conditional probabilities will distort the Poisson distribution. If constant shoot density is assumed, increases in shoot size will be translated into changes in projected shoot area based on the ratio of volume to projected area for the shoot. If, for example, it is

assumed that shoots are spherical in form, then the ratio of projected cluster area to projected leaf area will be

$$3\pi r^2/4\pi r^3 = 3/4r \quad (9)$$

where

r = the radius of the shoot.

Other geometrical forms will have more complex relationships, and projected area will change with sun angle and azimuth, but, regardless of the shape of the shoot or crown, equation 8 cannot be considered to be a valid model for foliage growth associated with shoot elongation or enlargement.

Spatial heterogeneity of foliage can influence both the quantity and pattern of light penetrating the canopy. Because clustering can occur at the level of the crown, shoot, or leaf, it is necessary to determine the extent to which heterogeneity at each of these levels affects light penetration before measured light below the canopy can be reliably used to estimate leaf area.

MODELING HIERARCHICAL EFFECTS IN SLASH PINE

Characteristics of Slash Pine

In slash pine, the carbon assimilation rate of individual needles is largely dependent on the light that those needles receive. Nearly 80% of the measured variance of CO_2 assimilation in needles can be accounted for by regressing assimilation against photosynthetically active radiation (PAR) (Figure 2).

The crowns of slash pine are conic, and the distribution of foliage is spatially limited to the volume defined by the crown form, which in turn is controlled by branch length (Figure 3). The canopy therefore is a collection of conic units each having its own individual needle distribution. Slash pines (and other pines) produce needles exclusively within the area of branch or stem wood created during the current season's growth. In slash pine, needle retention is less than 2 years (Gholz and Fisher 1982), and for this reason, needles are restricted to the branch area immediately adjacent to the actively growing apical meristems. This pattern of growth creates a non-

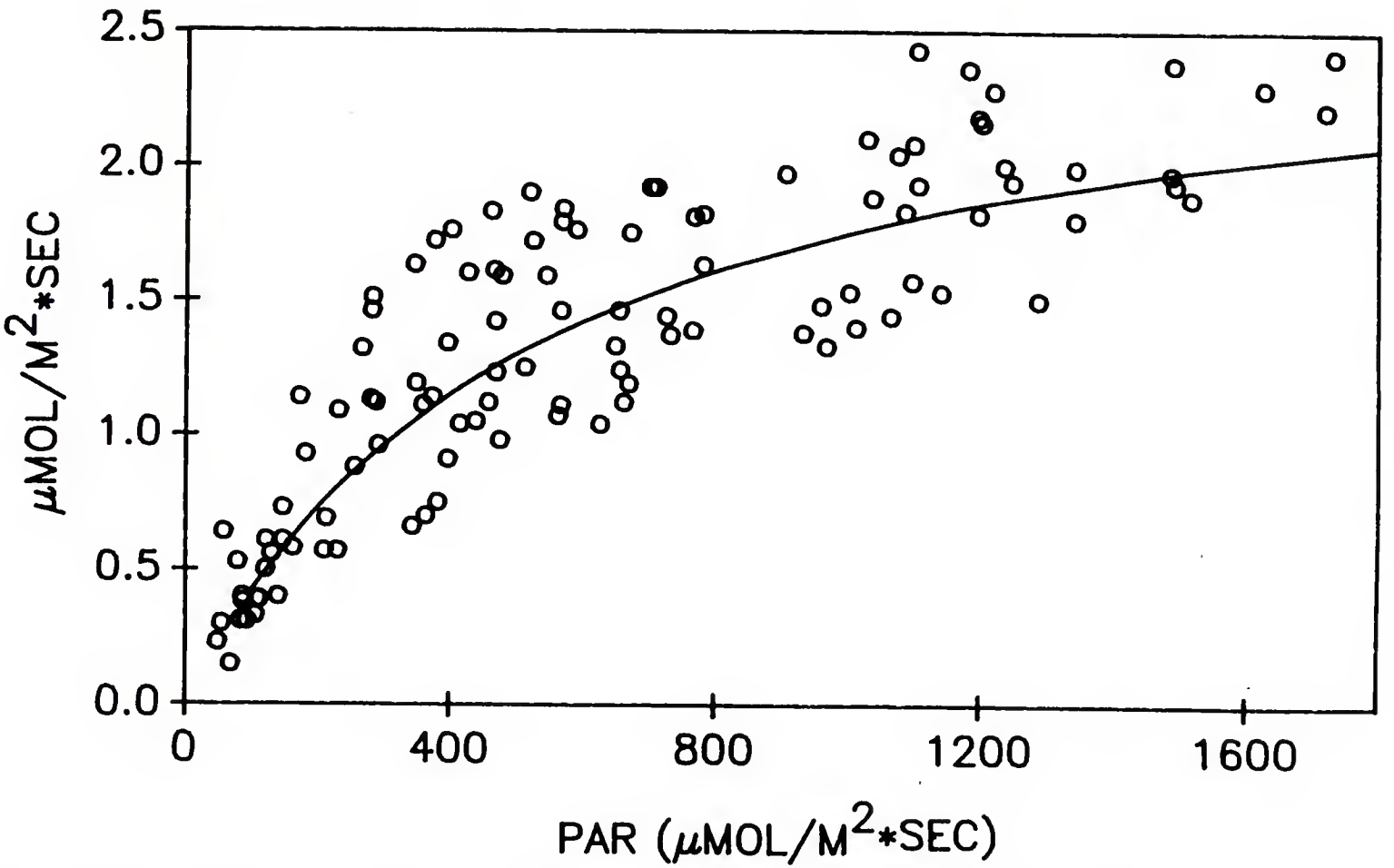

Figure 2: The effect of light on needle level assimilation rates. Each point represents the mean of 8 readings. (W. Cropper, personal communication).

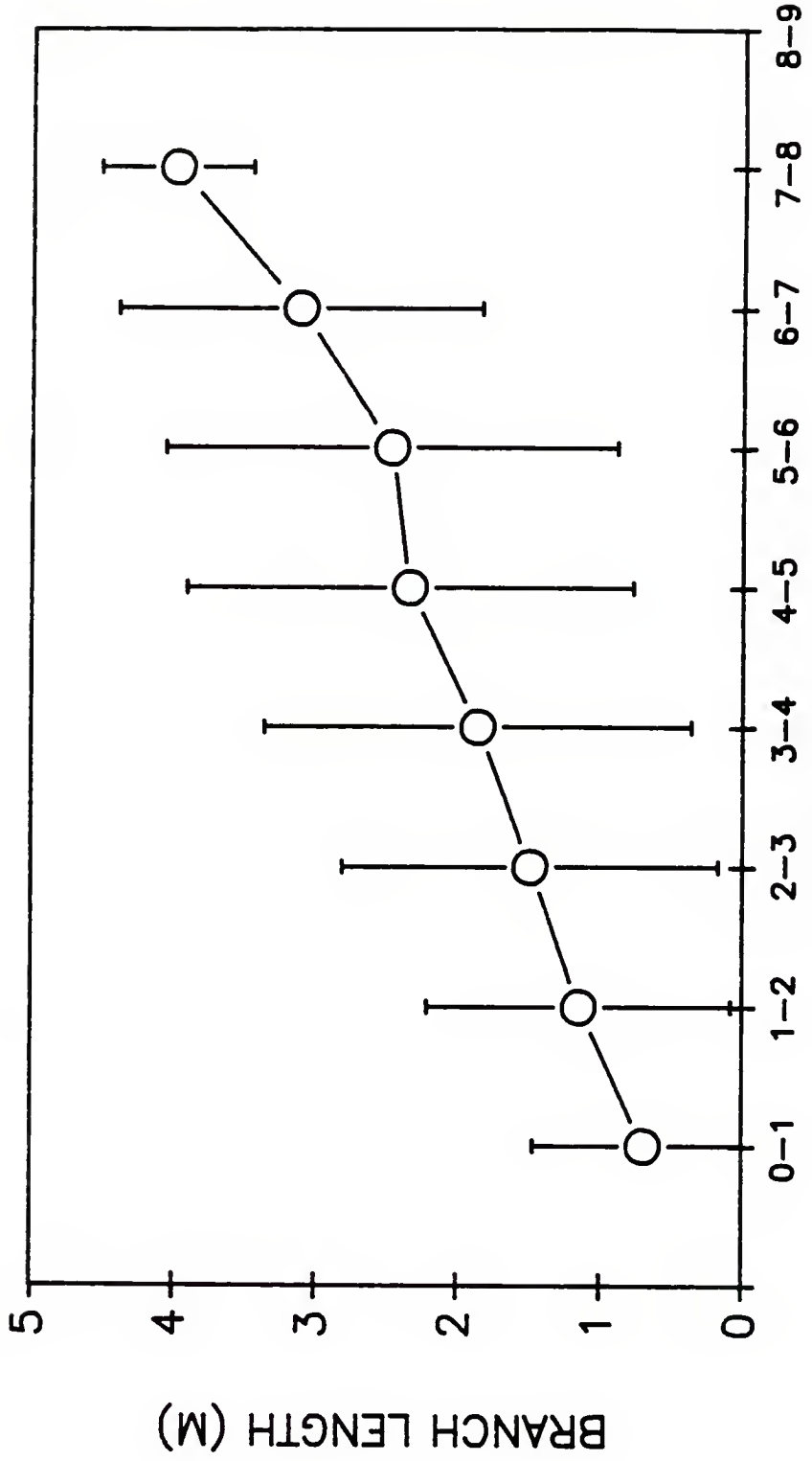

Figure 3: Distribution of mean branch length (± 2 standard deviations (SD)).through the canopy (n = 914).

foliated region surrounding the stem (Figure 4). The foliage distribution in a slash pine stand is therefore spatially heterogeneous.

The previous discussion of the impact of heterogeneity on light penetration suggests that light penetration through a slash pine canopy should be influenced by the canopy structure. Given the crown form and patterns of foliage growth, several hypotheses can be posed:

H1: Light penetration through a slash pine canopy differs in both pattern and quantity from light penetration through a homogeneous canopy.

H2: For canopies made up of crowns of discrete dimensions, differences in spacing pattern will have a stronger influence on light than is predicted for homogeneous canopies.

H3: If the crowns are arrayed in rows, the orientation of the rows will have a significant impact on light penetration and on the timing of maximum penetration through the stand.

Evaluation of Model Approaches

To explore the effects of various levels of heterogeneity on light penetration in slash pine, it is necessary to create a model that can simulate light

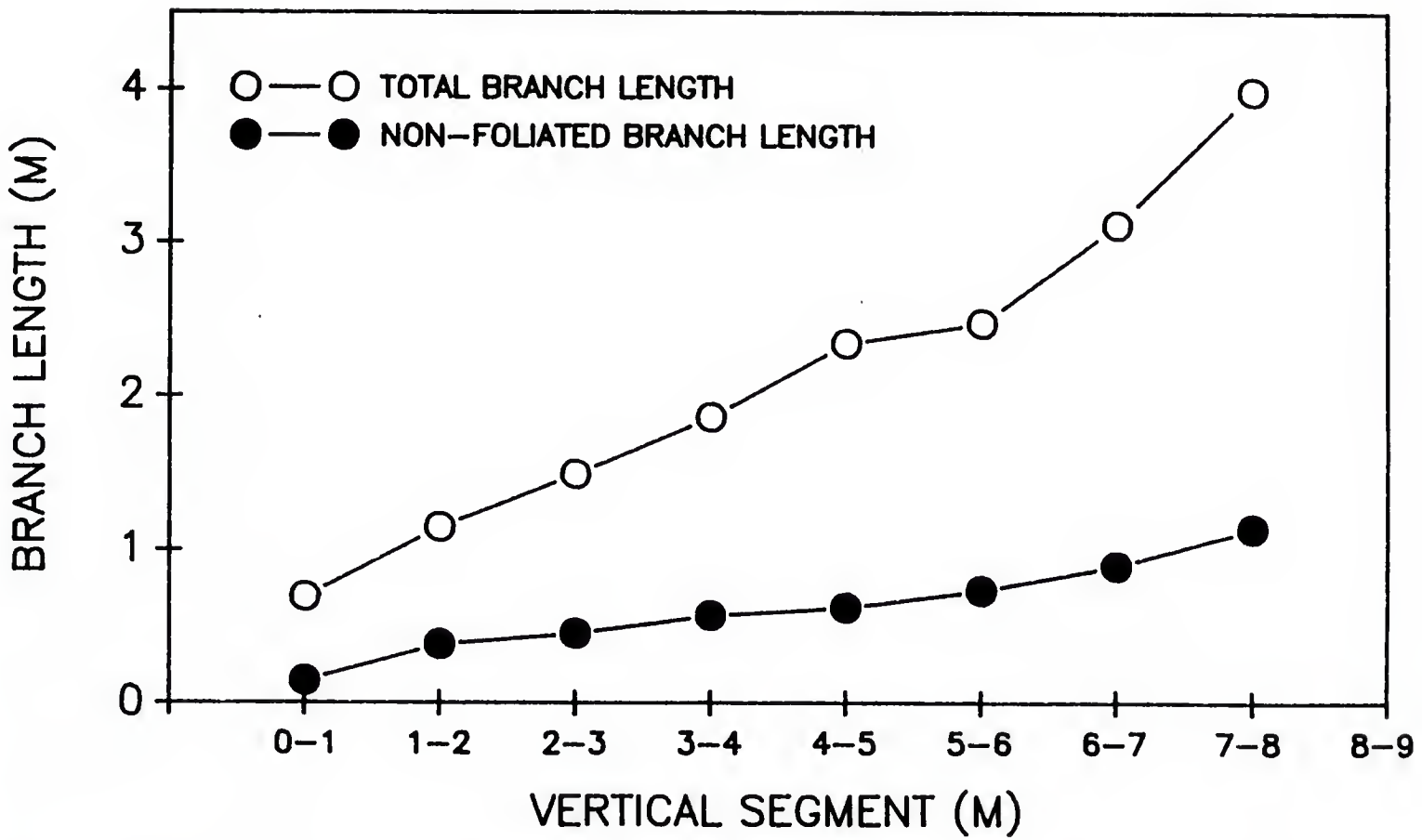

Figure 4: Distribution of average total and non-foliated branch length through the canopy. Non-foliated branch length is the length of the primary limb measured from the stem to the first foliated secondary limb.

penetration through a canopy with detail at both the crown and the cluster level. In the most common system for modeling discontinuous canopies, sub-canopies are represented as discrete geometric solids. In this type of modeling, the sub-canopies are usually defined at the level of individual plants (Mann et al., 1979, 1980; Norman and Welles, 1983; Rook et al. 1985; Whitfield, 1986; Grace, 1988) or sometimes hedgerows (Charles-Edwards and Thorpe, 1976; Cohen and Fuchs 1987). In this case, all the foliage of the plant or hedgerow is assumed to be confined within the volume of the solid, and equation 2 is generally assumed to be valid within this boundary. This type of modeling is called envelope modeling, because the leaf area of each sub-canopy is strictly confined to an 'envelope,' generally defined as an ellipsoid of rotation.

Envelope modeling is very flexible because each envelope can be given individual transmission properties, and envelopes can even be nested inside one another to simulate foliated and non-foliated portions of a plant (Norman and Welles, 1983).

In an array of canopies, the light reaching a given point can be described as

$$I_i = I_o[\exp(-k_1 z_1)*\exp(-k_2 z_2)*\ldots\exp(-k_n z_n)] \qquad (10)$$

where

$k_1 \ldots k_n$ = the extinction coefficients for sub-canopies 1...n, and

$z_1 \ldots z_n$ = the chordal distances defined by a beam of light as it passes through the array from a point source to a target.

Light penetration to a target surface is computed by calculating equation 10 repeatedly for a series of random locations on the surface and generating a statistical average (Norman and Welles, 1983).

To compute chordal distances efficiently, the solids must be geometrically simple. Asymmetries in crown form therefore present enormous computational difficulties. This simplicity of form in turn leads to a variety of model artifices that may be as severe as those of the simple models that they are designed to replace. Vertical density patterns are easy to model in a homogeneous canopy but are very difficult to model using envelopes. Because constant foliage density is generally assumed throughout each envelope, the vertical biomass profile of a crown is the cross-sectional area of the envelope as a function of height. For many plants, such as slash pine, foliage is concentrated toward the top and the outside of the crown volume. If this is the case, a constant density envelope will be artificially dense when viewed along its

longitudinal axis. (Generally plants are greater in height than in width.) To avoid this artifice it is necessary to nest envelopes with lower densities within each sub-canopy envelope. This does not, however, really remove the problem. Not only does nesting represent another level of model complexity, but the proper position and density of the internal envelopes must be based not on their actual attributes, but rather on the way in which they adjust the cross-sectional area of the outer envelope.

Envelope modeling can be used to model canopies at lower hierarchical levels such as individual shoots. Many of the problems associated with representing crowns as ellipsoids of rotation vanish when modeling is done at the shoot level. The effects of clumpiness on light penetration can be modeled, and crown form, foliage placement and transmission properties per shoot can be based on data. Assumptions of uniformity in size and density may also be more valid at the shoot level. Shoot level modeling also has the practical consequence of concentrating emphasis on a unit of vegetation that has properties that are measurable either in a laboratory or in a chamber in the field.

A great deal of current physiological work is done at the level of the branch, shoot and even the needle (the assimilation data presented here, for instance). If this

work is to be effectively scaled up to the crown and to the stand level, it is necessary to integrate these processes throughout the canopy. Ideally, light penetration should also be modeled for the same unit on which the physiological measurements are taking place.

Modeling a stand of slash pine at the shoot level, however, requires definition of the light regime at several hundred target surfaces per tree generated by light penetration through the several thousand potential shade objects that make up the surrounding stand. Furthermore, several runs will be needed to avoid artifices due to particular patterns of shoot placement. The problems associated with using an envelope model at the shoot level lie primarily in computational expense.

The modeling concepts discussed above compute light penetration through a canopy for light emanating from a single source. Diffuse radiation emanates simultaneously from all points in the sky dome. Computation of diffuse radiation involves integrating all the light from the entire sky:

$$P(X,Y,Z) = \frac{\int\int g(\theta,\delta)\ \exp(-kd\ S(\theta,\delta))\ \sin(\theta)\cos(\theta)\ d\theta d}{\int\int g(\theta,\delta)\ \sin(\theta)\cos(\theta)\ d\theta d} \quad (11)$$

(Norman and Welles 1983)

where

P = the probability that diffuse light penetrates to a point (X,Y,Z),

g = the light intensity function and, θ and δ are the angle of elevation and azimuth respectively,

d = the density of foliage, and

S = the sum of the z distances that a light beam would pass through if its point of origin lay along a vector starting at the point (X,Y,Z) with a direction defined by θ and δ.

Numerically, this equation is differenced into a series of direct light computations for a grid of sky locations. Computing diffuse radiation patterns for a single point with an integration step of 5 degrees therefore involves computing 1300 direct light readings.

Light Penetration Modeling in Discrete Space

Most of the computational burden in envelope modeling lies in the calculation of chordal distances through each sub-canopy. If, however, the sub-canopies are small and uniform in dimension and light penetration properties, then each sub-canopy can be treated as if it were particulate in nature. In this case, it is enough to know whether the sub-

canopy lies in the path of a beam. If it does, then it will remove a certain proportion of the light.

To calculate the presence or absence of a spherical sub-canopy in the path of a beam of light, a vector is drawn normal to the beam path from the beam to the center of the sub-canopy. If the magnitude of the vector is less than the radius of the sub-canopy, the beam will pass through the sub-canopy; otherwise it will not. This calculation is much simpler than computation of a chord. (It is the first step in computing a chord.) The calculation can, however, be further simplified if the coordinate system is itself aligned normal to the beam path, with the target point at the origin, and scaled so that one unit equals the radius of the sub-canopy. The presence or absence of the sub-canopy in the beam path can then be determined by rounding the center of the sub-canopy to the nearest integer node. If, after rounding, the center of the sub-canopy has coordinates of (0,0) in the dimensions normal to the beam path and a magnitude greater than 0 in the dimension parallel to the beam, the sub-canopy will intersect the beam. If all of the sub-canopies are the same size, then the entire system can be scaled and rounded simultaneously, and all of the sub-canopies that intersect the beam can be located at the same time. In this approach, the sub-canopies are positioned in continuous

space, but light penetration is computed in discrete integer space. If the target surface itself is composed of a field of equal-sized sub-canopy units, then the entire system, including the points representing the target, can simultaneously be scaled and rounded. The light penetration to the entire target can then be determined through a series of row searches.

In discrete space, sub-canopies lose explicit dimension. The dimension is only reflected indirectly in the scaling factor. Sub-canopies become points with light penetration properties defined as the average light penetration per unit area of the sub-canopy. The targets (and therefore the light beams) are also restricted to integer space, and light penetration to a target point represents the average light penetration to a grid-sized area surrounding the target. This discrete approximation keeps the fundamental logic of the envelope model, modeling the system as a series of sub-canopies, but allows its application on a much finer scale or on slower computers due to the decrease in computational burden. Using this technique, modeling on a cluster level can be accomplished on micro-computers.

In discrete space, a homogeneous medium of unit thickness is represented as a plane of filled nodes. As this system is rotated in relation to a light source, the

discrete nature of space confines the nodes to integer locations, causing raster effects similar to those seen in low-resolution computer graphics. A plane diagonal to the light source will be represented in the model as a series of smaller planes offset from one another by one node. If it is assumed that a Beer's law model is appropriate for the continuous system, then, as the relative sun angle shifts, the light penetration through the system will decrease as a function of the relative angle:

$$I_i = I_o \exp(kdz/\sin\theta) \qquad (12)$$

where

θ = the relative angle between the plane and the light source.

If kdz is constant for all θ, then equation 12 simplifies to

$$I_i = I_o \exp(k_f/\sin\theta) \qquad (13)$$

where

k_f = a collapsed constant describing light penetration normal to the plane.

If k_f = 1 and the relative angle is 45 degrees, then $1/\sin\theta$ = 1.41 and I_i is predicted to be 0.243 I_o.

If this same system is modeled as a discrete entity, half the points will be shaded by two points and half by one, leading to a predicted I_i of $0.252*I_o$. In this case the prediction error is less than 1%. Because the exponential decline function is uniformly convex, the discrete model will always predict light penetration greater than or equal to a continuous model. The divergence is, however, not large. As the grid size decreases, the system converges on the continuous representation.

Modeling light penetration into a stand using the discrete cluster level model involves five steps:

1.) Placing the clusters into space and assigning light penetration properties to each cluster.
2.) Rotating the shoots to align the system with the sun position.
3.) Rounding the clusters to the nearest integer node.
4.) Searching for the presence of clusters between the target nodes and the sun.
5.) Quantifying the light attenuation caused by all shoots by matching clusters with their specific light attenuation properties.

To summarize, discontinuous canopies can be modeled by creating geometric representations of the stand and simulating actual light penetration. This approach has traditionally been accomplished by modeling crowns as

elipsoids in real space. For modeling at the shoot level, however, this approach is extremely inefficient. Discrete space models offer a more efficient alternative and therefore are a preferable approach for shoot level modeling.

A DISCRETE-SPACE MODEL FOR SLASH PINE

Data Acquisition

In this study, a model was based on a stand of plantation-grown slash pine. The stand age was approximately 20 years. Total tree height was approximately 17 m and the crown depth was approximately 7 m. Stocking density was approximately 1200 stems*ha^{-1}.

The parameters used to model the location of foliage in space were based on a data set derived by destructively sampling 40 trees in August 1987 and 60 trees in August 1988. Pertinent data obtained through sampling included the number of branches, branch position, branch angle, total branch length, non-foliated branch length (length to the first live side limb), number of needle clusters (shoots) per branch, and biomass per branch.

One large and one small branch were removed from each 1 m segment down through the crown. The top segment began at the base of the terminal and segments were marked down the stem to the base of the crown. Crown depth varied between 3 m and 9 m. A small limb was only measured if two

clearly defined classes of branches existed within the segment. If all the limbs were the same size, the limbs were all classified as large, and only one limb from the segment was sampled. Large and small were therefore relative classifications. The point at the base of the terminal leader where the top segment began was classified as the top of the tree. The terminal, which represented the current year's growth, was treated as a branch and was not sampled. For each limb sampled, the position of the base of the limb and the tip of the limb relative to the tree top were recorded. From these data branch angles were determined using the following formula:

$$AOE_b = \arcsin[(D_{top} - D_{base})/L_t] \qquad (14)$$

where

AOE_b = the branch angle from the horizontal,

D_{top} = the distance from the tip of the branch to the top of the tree,

D_{base} = the distance from the base of the branch to the top of the tree.

L_t = the total length of the branch.

The bias due to sweep in the limbs was minor and was ignored.

Foliage position was determined by assuming that foliage was restricted to the region from the tip of the branch to the first foliated bifurcation of the main branch. Within this region, foliage was assumed to occur with equal probability, and the needle biomass was therefore uniformly distributed. Using this technique, foliage was added to a standardized stem composed of 1 m segments to build up a vertical foliage profile for the stand.

Because the branches were angled between 50 and 70 degrees toward the vertical (Figure 5), correcting for branch angle shifted the calculated position of the foliage toward the top of the canopy (Figure 6). For a 2 m long branch with a non-foliated length of 0.5 m, a D_{base} location 2.5 m below the top of the tree and an AOE_b of 50 degrees, this technique would allocate 3% of the foliage to the 0 to 1 m segment, 86% of the foliage to the 1 to 2 m segment, and only 10% to the 2 to 3 m segment where the branch originated. Even after adjustment, however, the foliage is probably placed lower in the crown than its actual position. This is because foliage is assumed to be evenly distributed within the foliated region of the limb, when, in fact, most of the foliage is concentrated toward the branch tip. A uniform distribution pattern was used, because measuring the actual distribution of foliage in

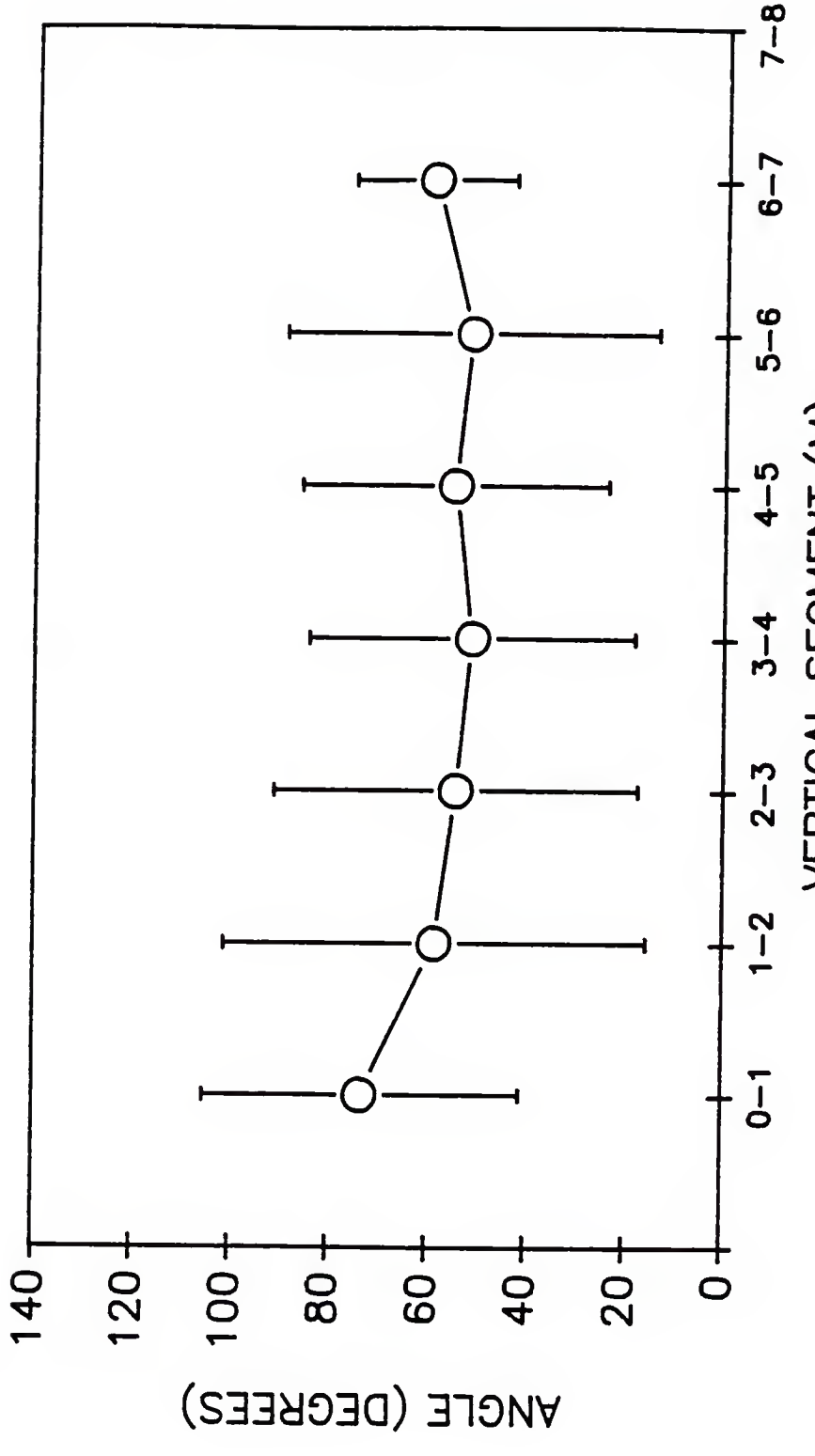

Figure 5: Distribution of branch angle (± 2 SD) through the canopy.

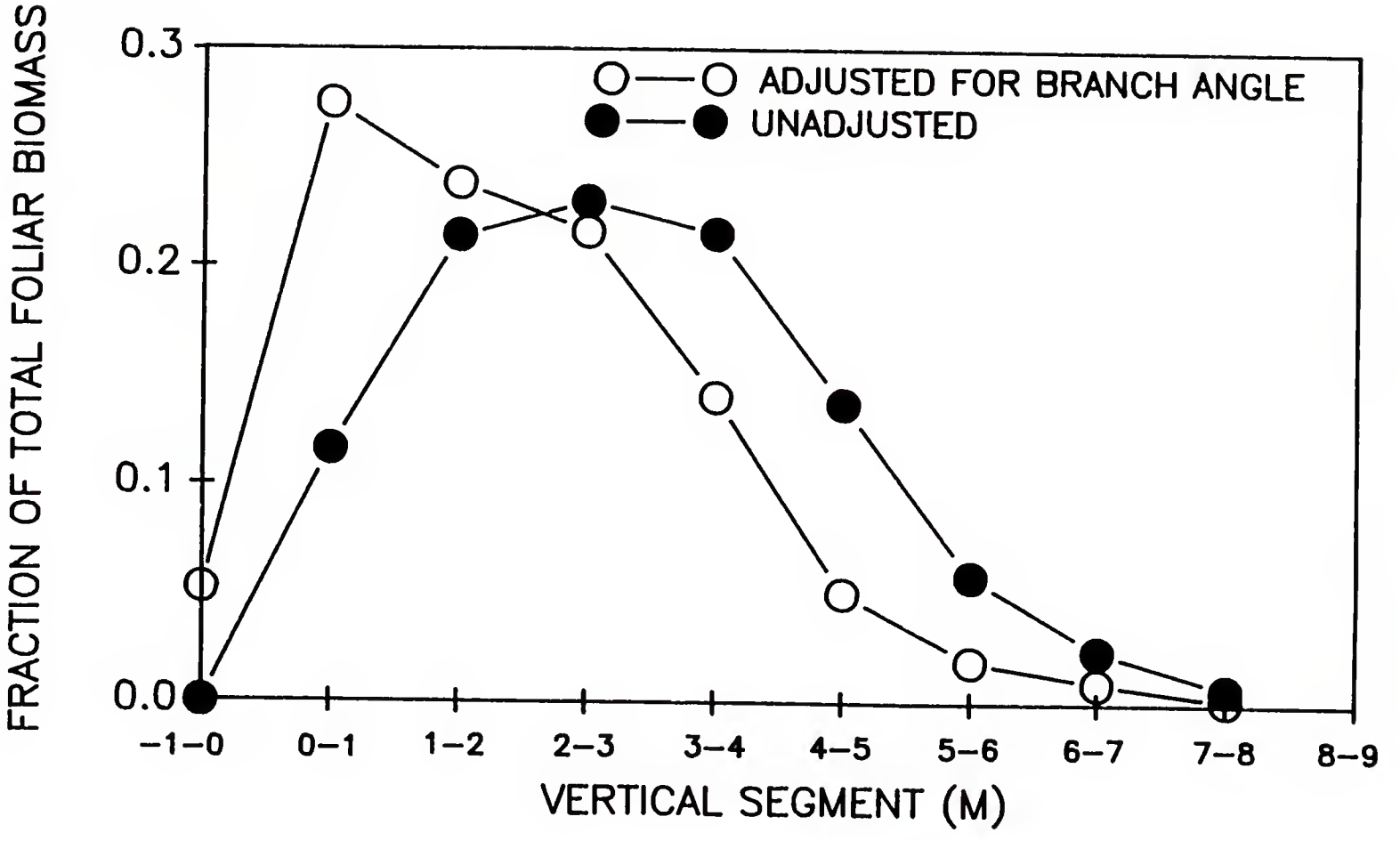

Figure 6: Estimated foliage profiles. The adjusted profile is derived by shifting foliage to compensate for branch angle.

space was not possible due to the breakage that inevitably accompanied the felling of trees.

For each limb, the number of needle clusters (apical growth meristems) was counted. The system used to distribute foliage was also used to distribute clusters, with the number of clusters per limb substituted for the dry weight biomass. Cluster distribution has a random component, because the number of clusters is discrete, and clusters are placed with equal probability along the foliated limb rather than in equal proportion.

The horizontal distribution of foliage around the stem was also determined using the branch angle and the foliated region of the limb. The distance of the foliage from the stem was determined by the cosine of the branch angle rather than its sine. The horizontal distribution was assumed to be uniform in regard to azimuth.

In the model, two techniques can be used to place foliage in space. One, which was used in a preliminary model (McKelvey, 1988), used the vertical distribution of biomass. In this approach, needle clusters were assumed to contain an equal quantity of biomass. Needle cluster position was therefore derived from biomass data, and clusters were placed in vertical space by a weighted random process based on the biomass distribution. Horizontal

placement was based on the branch length at the chosen height.

The second system for distributing needle clusters uses the vertical distribution of the clusters. In this approach, foliage biomass enters indirectly and is used to determine the extinction coefficient for each cluster.

This second method was used in the model presented here because it places the emphasis on positioning the clusters accurately in space. The previous discussion of heterogeneity suggests that changes in the position of foliage should have more impact on light penetration than shifts in biomass per cluster. The slope of the exponential decline curve is steepest for values near 1 and progressively flatter as values approach 0. In a sub-canopy model, where there are gaps in the foliage, the model should therefore be most sensitive to the size of the gaps than to the properties of the foliage filling the area between the gaps. For purposes of crown construction, the rules for placing clusters in space create the gap structure, and the properties of the clusters define the area between the gaps. The priorities for modeling should therefore primarily be to position the needle clusters accurately in space.

To determine light penetration through needle clusters within the canopy, it is necessary to determine whether

direct light penetration is related to needle biomass per cluster, and whether the orientation of the cluster to the light source affects either penetration or projected area.

To obtain these data, a device was designed to allow controlled directional light to be directed onto clusters from a variety of angles (Figure 7). The device consisted of a light source, a collimation tube and a light sensor which was held in a fixed position relative to the light source. A 4 cm parabolic mirror focused the light on a Li-Cor PAR spot sensor, eliminating angle coverage effects due to large objects such as branches. Clusters were placed between the sensor and the tube. The light source was mounted on a stand with slip joints that allowed a full range of movement in three dimensions. This allowed the cluster to remain fixed and the light source to be moved, allowing the cluster to remain in an orientation similar to its natural orientation on the tree. The cluster was held in a movable stand (Figure 8). By moving the cluster vertically and horizontally through a series of fixed positions it was possible to define a grid normal to the light source for any sun angle. By rotating the cluster position in the clamp it was possible to simulate any azimuth.

There is little variation in the architecture of needle clusters. Needles radiate from a central limb and are

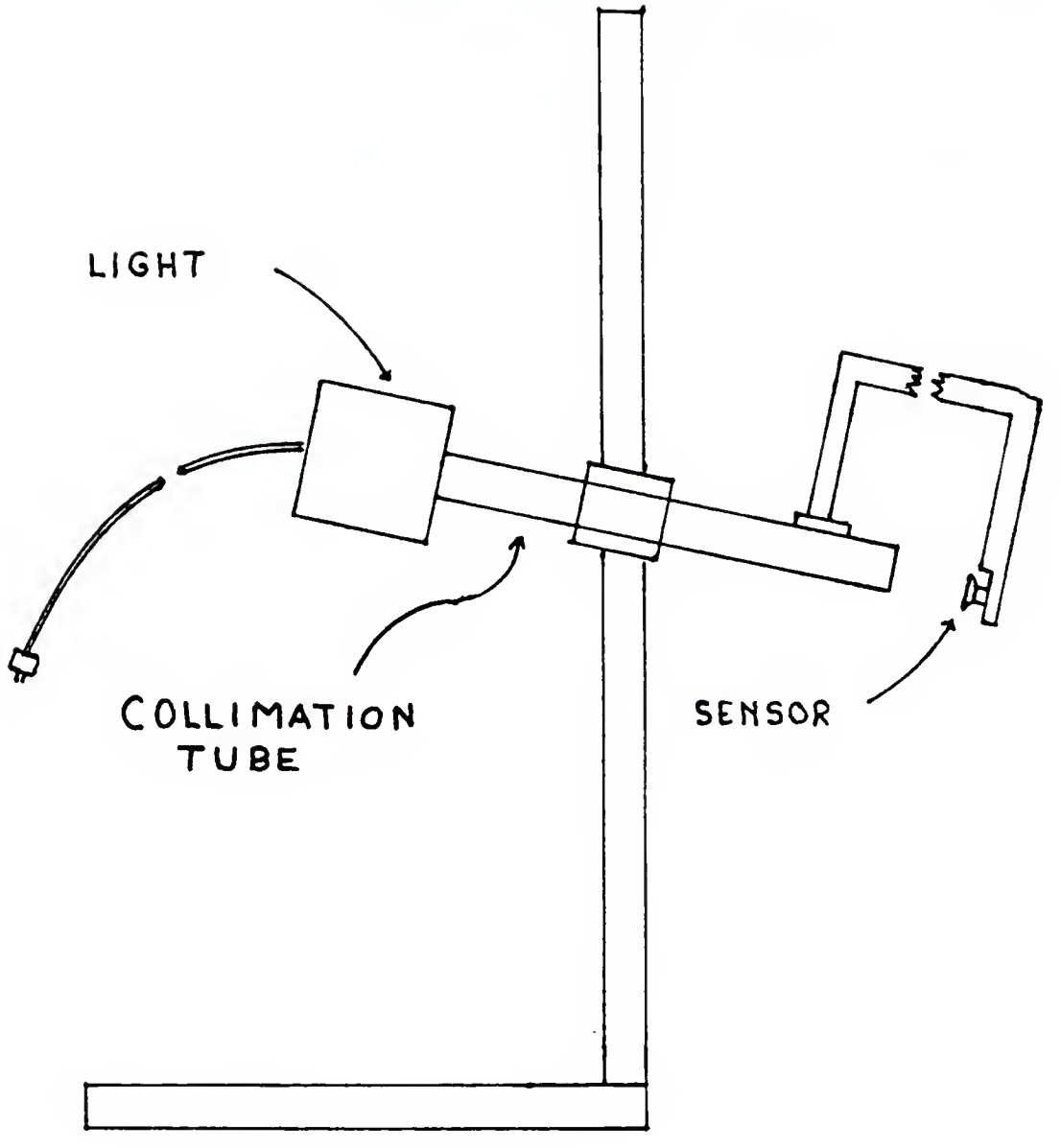

Figure 7: The device used to determine light penetration through clusters.

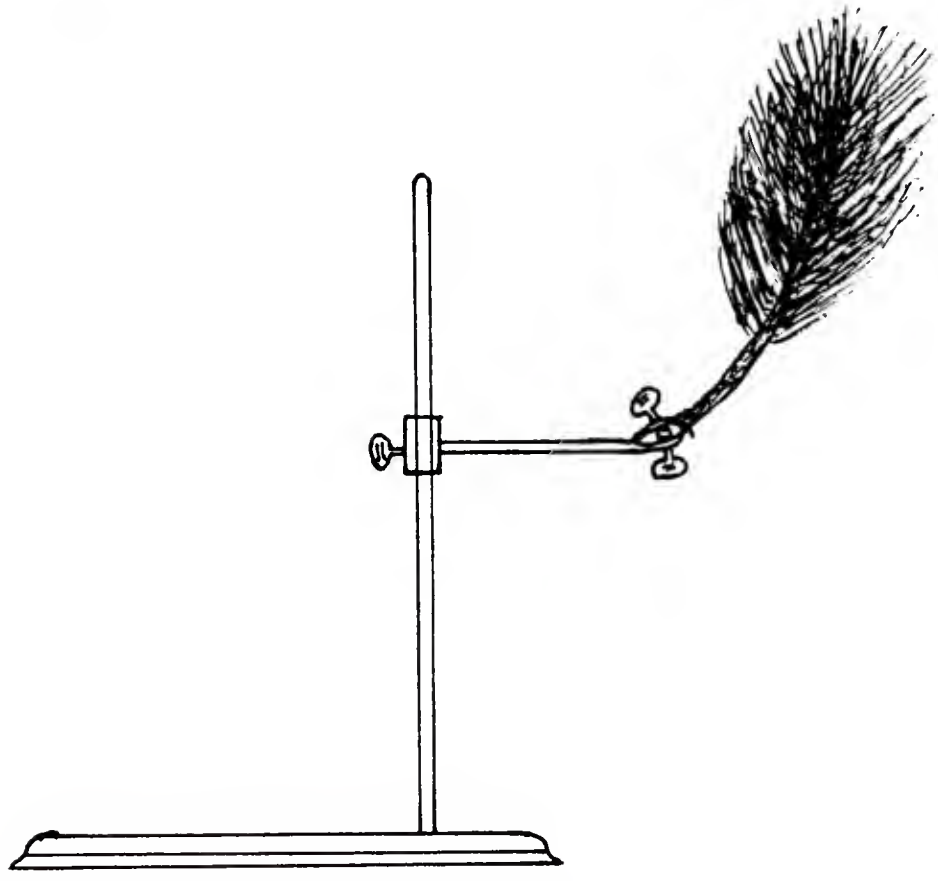

Figure 8: Apparatus for positioning needle clusters for light penetration measurements. Vertical and horizontal movement of the stand, coupled with angular movement of the light source ,allows any angle and azimuth to be reproduced.

generally angled toward the branch tip. When viewed from the branch tip, clusters are circular, with a dense center defined by the apical meristem and a maximum radius equal to the needle length. In this orientation, clusters are circular with a radius of 28-35 cm regardless of biomass. From the side view, clusters are less similar in form. The length of the cluster is a function of the growth rate over the past season or two, depending on the time of the year. In April, the cluster holds one cohort of needles, and in October it holds two. In general, because new needles grow at a very acute angle from the stem, the region beyond the tip of the limb forms a hemisphere defined by the needle length. Below the hemispherical end, there is a portion of indeterminate length that is cylindrical. At its base, this cylinder can terminate in a hemisphere if the needles are old and droop, or in a triangle if the needles are young and are still stiff and angled toward the branch tip.

Light penetration and projected cluster area will diverge to the greatest extent when the cluster is viewed from the end (parallel to the limb) relative to the side (normal to the limb). Light penetration through the cluster from all other angles will encounter a form intermediate between these extremes. Measuring light penetration parallel and perpendicular to the limb therefore brackets

the range of penetration and area projection from all potential sun angles.

Needle clusters were sampled at 5 cm intervals over a grid of 30x30 cm. With the 2 cm overlap caused by the parabolic mirror, the area covered was approximately 34x34 cm. This area is large enough to encompass the end views of all needle clusters in their entirety. It encompassed the side views of smaller clusters, but larger clusters (up to 52 cm long) projected beyond the grid. Sample clusters were centered on the grid using the apical meristem as the center for end views and the midpoint of the cluster for side views. Light penetration was calculated for ten clusters from the end and the side and the results compared. The average light penetration values per unit area for these two views differed by less than 10% among all clusters (average = 5.1% less light penetration from the side). Rounded to the nearest 10% and compared using a chi-square test of fit (Table 1, Figure 9), six clusters showed no significant difference ($p < 0.05$) and four showed significant differences in pattern. Significant differences between the end and side views were largely due to an increase in projected branch area when viewed from the side.

Small needle clusters do not change dimension or light penetration properties when the relative direction of the

Table 1

Comparison of light penetration through clusters viewed from the end and from the side. Light penetration is the average fraction of light penetrating a cluster. Clusters marked with an asterisk were significantly different ($p < 0.05$) in pattern based on a chi-square (X) text of fit. The X 0.05 critical point with 8 degrees of freedom = 15.5.

			Light Penetration (%)			
Cluster	Width (cm)	Length (cm)	End view	Side view	Difference	X
1	35	46	57	61	4	5.2
2	25	26	87	86	1	8.2
3	30	32	87	87	0	9.7
4	30	48	63	54	9	23.6 *
5	32	32	89	89	0	4.1
6	32	40	68	62	6	25.3 *
7	32	42	72	62	10	17.7 *
8	28	50	72	62	10	51.2 *
9	28	28	84	77	7	11.5
10	22	20	97	93	4	2.2

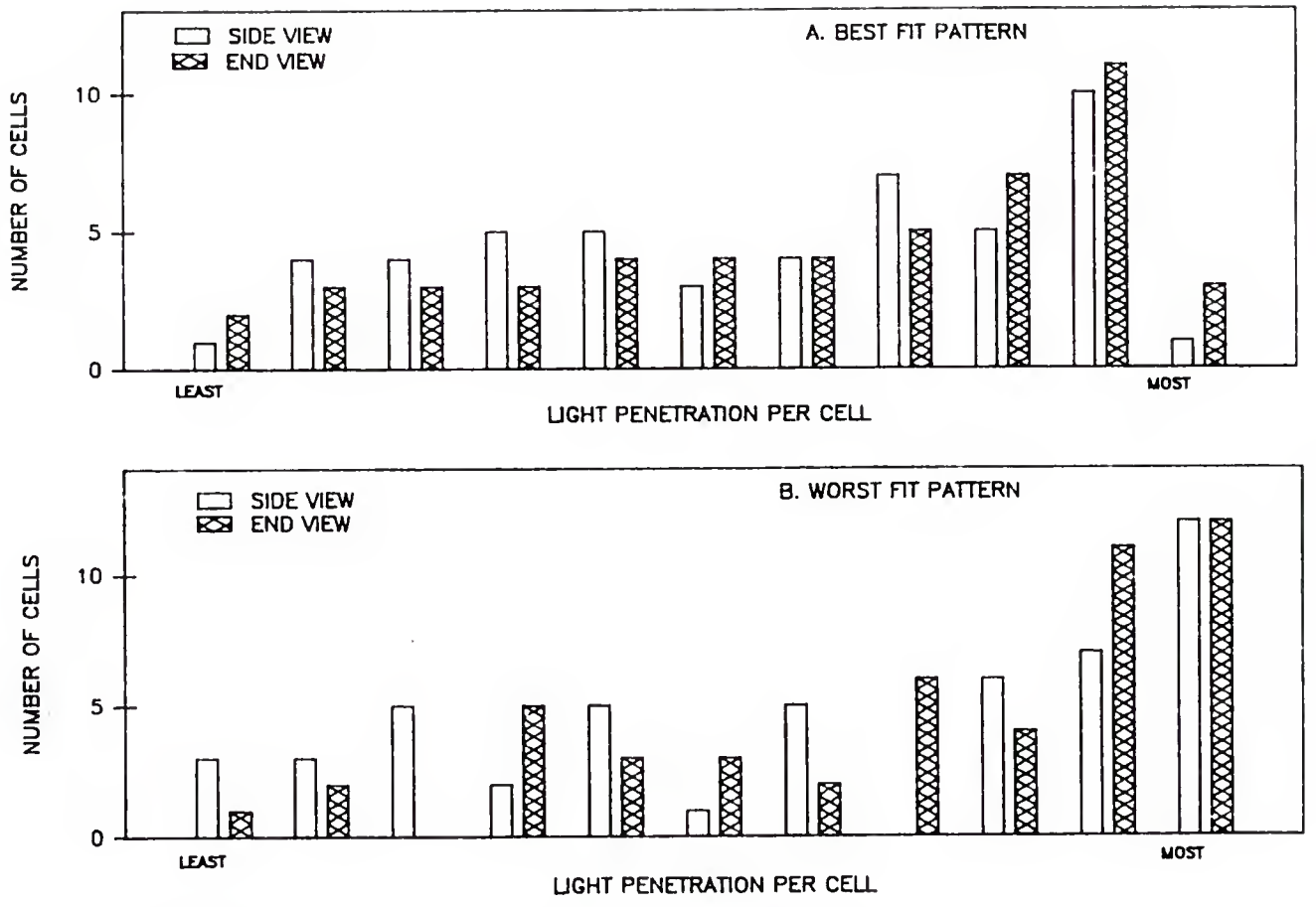

Figure 9: Patterns of light penetration through needle clusters when viewed from the side and from the end. A cell is the 5x5 cm area represented by an individual light penetration reading. The best fit and worst fit patterns correspond to clusters 4 and 8 in Table 1

light source changes. The larger clusters are similar in density, but will project as much as twice the area when viewed from the side rather than from the end. This can lead to a disparity between modeled area and actual projected area if these clusters represent a significant proportion of the total cluster population.

To determine the proportion of proportion of large clusters in the canopy, the lengths of all of the clusters were measured on five trees with stem diameters ranging from 15.8 cm to 25.2 cm in diameter at breast height (dbh = 1.37 m). Of the 1585 clusters sampled, 73% were less than 33 cm in length and 89% were less than 40 cm in length (Figure 10). Based on this sample, the maximum error in estimated cluster area due to the presence of large clusters was less than 7%. At minimum biomass levels, the canopy is primarily composed of clusters small enough to project equivalent area and density from all sun angles. The assumption that clusters can be treated as points with average area and uniform penetration properties, a necessary condition for discrete cluster-level modeling, is therefore justified.

A regression for cluster penetration as a function of biomass was constructed. For the regression, light penetration parallel to the limb was measured for an additional 10 clusters. The end view was chosen because it

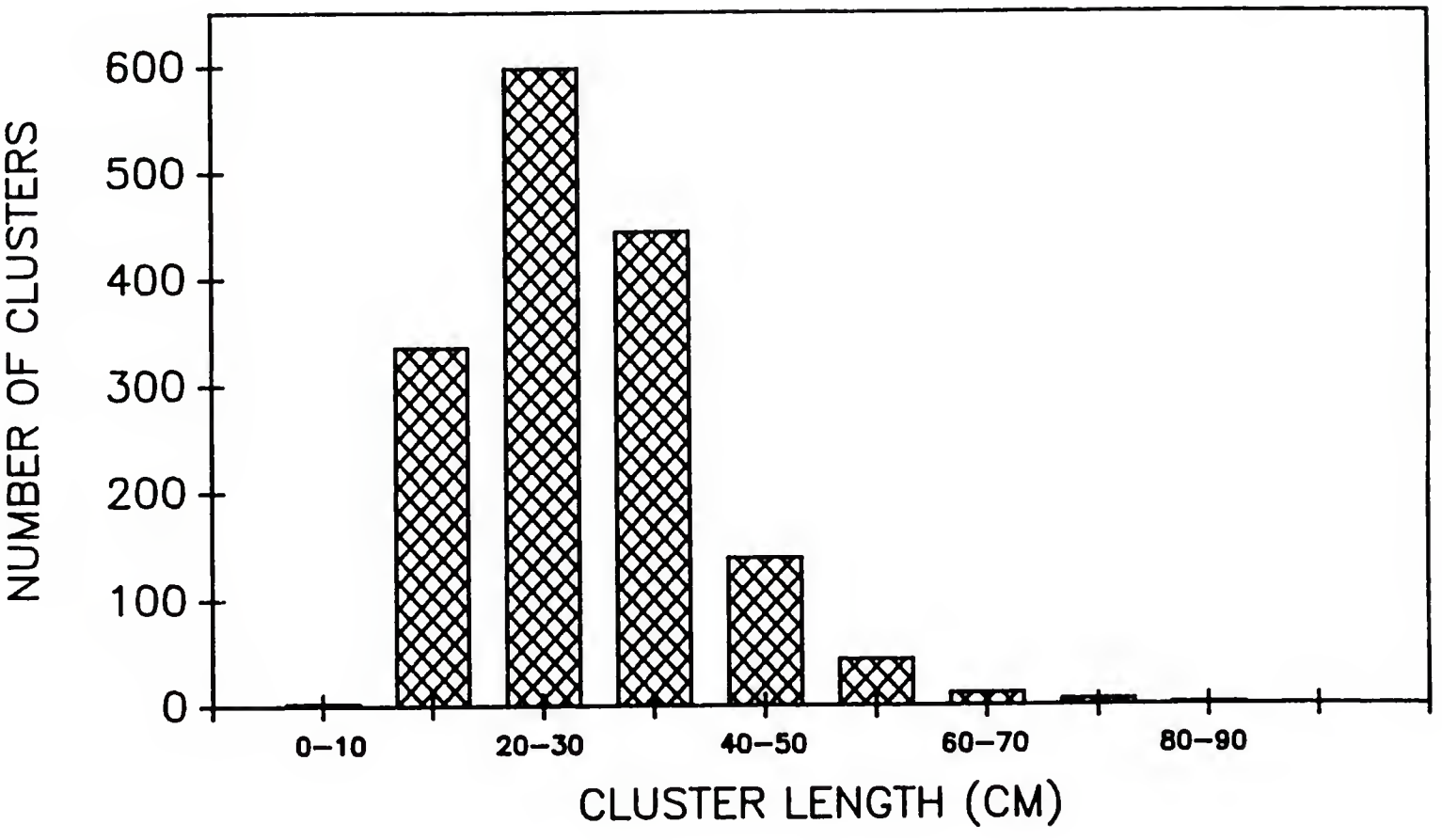

Figure 10: The frequency distribution of cluster length in the canopy.

captures all the foliage regardless of cluster length, and because the orientation of the clusters on the tree suggests that the end view dominates. (If clusters grow toward light, the limb, which represents the change in position of the meristem over time, must lie roughly parallel to the beams of light emanating from the light source.) A log-linear (exponential decline) curve, was chosen for the fit (Figure 11) because it is of the same form as the exponential decline curve and the variables are analogous to those used in Beer's law modeling. The curve also has stable properties when used to extrapolate beyond the edge of the data set.

Model Structure

Cluster level modeling is very flexible, and a variety of configurations can be used to model the stand. For all of the results presented here, the stand was modeled using the following rules:

1.) All trees were built using the same procedure (Figure 12). The foliage quantity, cluster numbers, branch angles, and branch lengths were identical for all trees. The parameters were the mean values derived through the destructive sampling. Within the bounds of this rule set, the actual placement

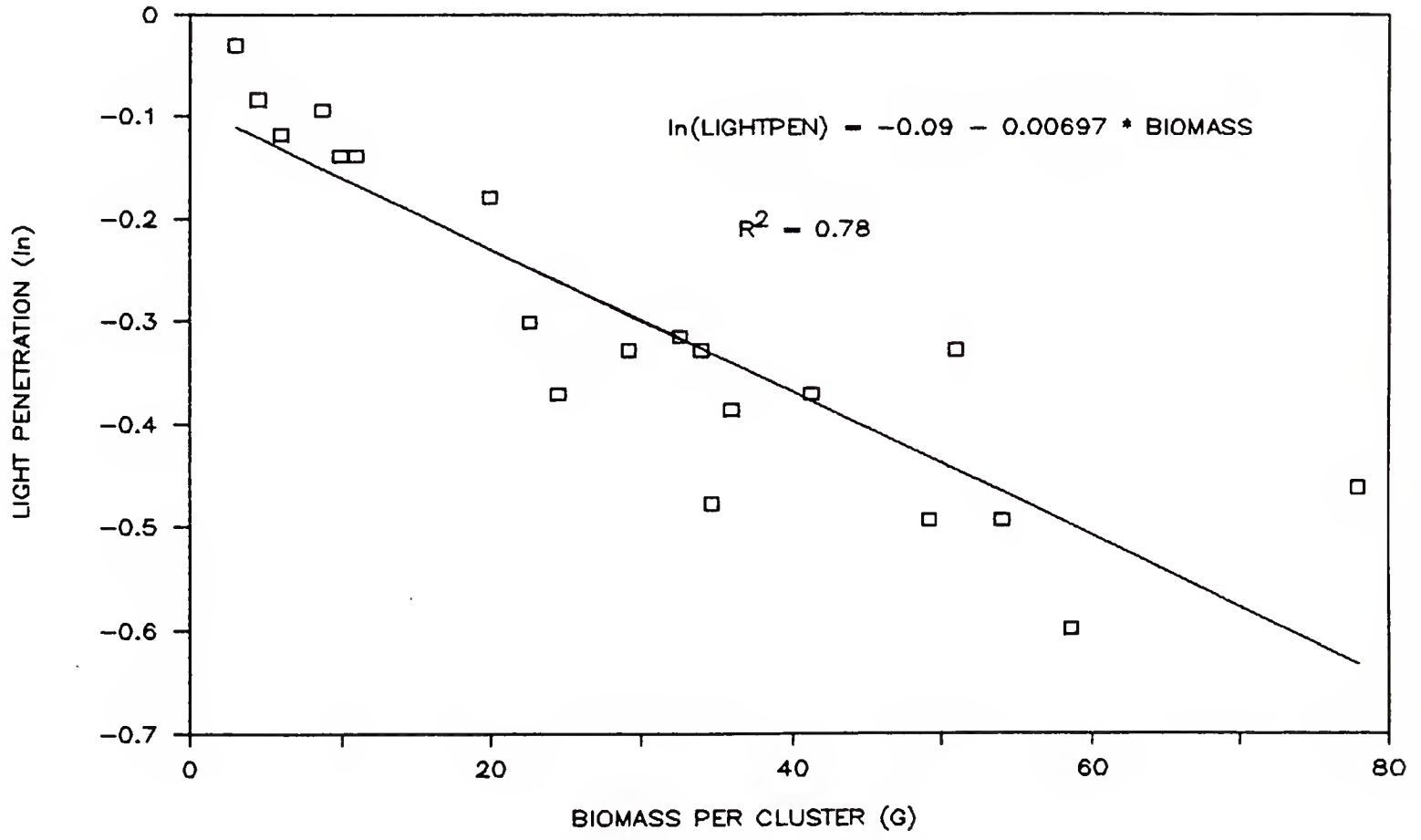

Figure 11: The pattern of light penetration through needle clusters as a function of all-sided LAI.

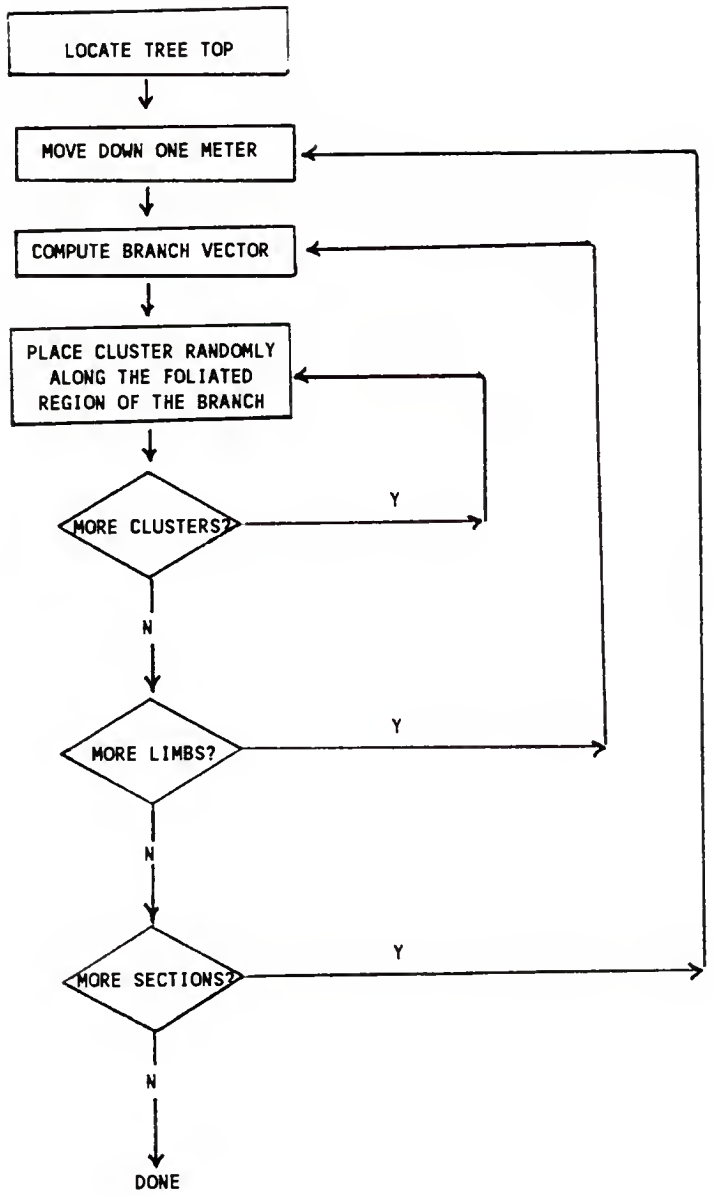

Figure 12: The rules for building a tree in the cluster level model.

of clusters in both azimuth and angle was random.

2.) All the trees were identical in height.

3.) The trees were evenly spaced on a grid aligned with the cardinal compass directions. Rows and columns ran exactly north-south and east-west, respectively.

In the model, the clusters are held as locations in three-dimensional space. Light penetrates the system of clusters obliquely as a series of parallel beams. To determine light penetration through the system, the discrete approximation technique outlined above requires that the coordinate system be aligned normal to the light source before scaling and rounding. The alignment is accomplished by rotating the cluster positions around an arbitrary origin until they occupy the positions in the space that they would have if they had remained fixed and the cartesian coordinate system had been rotated. The process starts with an initial orientation in which the X and Z axes define a plane parallel to the ground and the Y axis is height. The initial sun location is at 0 degrees AOE and parallel to the X axis. Cluster rotation is always relative to this initial configuration and is inverse to the movement of the light source. If the light source moves up and to the right, the cluster positions are rotated down and to the left. Angles of rotation required to align the

axes with daily sun angles and azimuths can be generated through trigonometric relationships, and equations are available in the literature (Campbell, 1977; Walraven, 1978; Gates, 1980).

Rotation of points around an arbitrary origin requires several transformations. The point set must first be centered on the origin, then two scalar rotations are necessary. The first aligns the axes with the sun azimuth, the second with the AOE. This process can be accomplished efficiently through the use of transition matrices.

Any rotation of a group of points in Cartesian space can be viewed as a series of individual translations of coordinate vectors, each vector representing an individual point. The rotation algorithm can therefore be determined by tracing the translation of a single coordinate vector, C, through multiplication with a series of transition matrices, $T_1, T_2, \ldots T_n$. The first translation can be written as $C' = C * T_1$, the second $C'' = C' * T_2$ and so forth. $C * T_1$ can, however, be substituted in place of C', and the second translation can therefore be expressed as $C'' = C * T_1 * T_2$. This process can be continued for as many transition steps as are necessary to accomplish the desired rotation. A single transition matrix T_f is created which is the product of matrices $T_1 \ldots T_n$. A single vector - matrix multiplication is then required to rotate each coordinate

vector to its final position. The rotation process therefore always begins with the same set of initial coordinates, and there is no cumulative buildup of errors during rotation. After rotation, each coordinate vector is multiplied by a scaling constant and rounded to the nearest integer location. If the point is equidistant between two integers, the coordinate is rounded to the even integer.

Once the clusters have been rotated and their positions rounded, the amount of light penetrating to the target can be computed. If all the clusters have equal transmission properties, light penetration to any point can be computed as

$$I_i = I_o \exp(-kn) \qquad (15)$$

where

n = the number of clusters.

If the transmission properties are different, then light penetration is computed as

$$I_i = I_o \exp[-(k_1d_1 + k_2d_2 + \ldots k_nd_n)] \qquad (16)$$

where

$k_1 \ldots k_n$ = the extinction coefficients for each

individual needle cluster, and

$d_1 \ldots d_n$ = the densities.

If the light penetration per cluster is known (rather than computed through a Beer's law relationship), then this calculation collapses to a product series:

$$I_i = I_o \prod_{i=1}^{n} P_i \qquad (17)$$

or, if the penetration is identical for all clusters:

$$I_i = I_o P^n \qquad (18)$$

With the Cartesian coordinate system parallel to the solar radiation along the X axis, the task of searching for shade becomes the search for points that have the same Y and Z coordinates as the target and larger X values. This search can be accomplished efficiently by first sorting the potential shade clusters according to either the Y or Z coordinate values. The array can then be searched progressively. In certain cases it may be beneficial to sort first by the X coordinate and to clip those points with X values less than the target, but because the target is generally placed so that the maximum potential number of shade clusters lie between it and the sun, this sort is not coded in the Appendix.

The model was run for angles and azimuths representing one quarter of the sky dome. Light penetration was simulated for AOE from 30 to 85 degrees and azimuths of 185 to 270 degrees at 5-degree intervals. For below-canopy readings, the target was a 5x5x1 m rectangular region below the canopy containing 196 measurement points set at random locations. For measurements within the canopy, two target arrangements were used. To simulate a generalized light penetration profile, a series of 5x5x1 m planar targets were placed at 1 m intervals within the canopy. To simulate light penetration to the cluster surfaces, a target tree was used with each cluster acting as a measurement point. The vertical position of each cluster was used to determine a light penetration profile to the surface of the clusters.

This modeling approach, in an evenly planted stand, allows simulation from all potential angles (above 30 degrees) and azimuths. The other three quarters of the sky dome can be filled in by assuming symmetry. The minimum sun angle that can be modeled is dictated by the distance from a target to the edge of the simulated stand and the depth of the canopy. The question is whether a tree that is not modeled has the potential to influence the light environment at a target point. For a field of conic crowns and a target at the base of the canopy, the distance to the

edge of the system necessary to ensure that edge effects do not affect light penetration to the target is

$$D_e = \tan(AOE)/z_c \qquad (19)$$

where

D_e = the distance to the edge of the system, and

z_c = the depth of the canopy.

This function becomes infinite as AOE approaches 0 and breaks rather sharply below 30 degrees AOE. In Florida, direct light emanating from AOE greater than 30 degrees dominates the daily light integral. Even at the winter solstice, the sun is higher than 30 degrees for a 5-hour period in the mid-day. For Florida, 30 degrees is therefore a reasonable minimum angle. For more northern latitudes, much of the light may emanate from AOE less than 30 degrees.

Model Results

In a three dimensional model, asymmetries in foliage distribution may make light penetration strongly dependent on azimuth. In a row crop, the highest penetration through the canopy is expected when the sun azimuth is parallel to the rows. In the simulation, at low AOE, more light

penetrates from an azimuth of 270 degrees (Figure 13). In the modeled tree configuration, an azimuth of 270 degrees is exactly parallel to the rows. This phenomenon is, however, only noticeable at low AOE and only occurs if the sun azimuth is precisely aligned with the row. An Azimuth of 185 degrees, for instance, is only 5 degrees out of line with the row orientation. Azimuth is therefore not a significant factor in below-canopy light penetration. This means that light penetration curves can be created based on sun angle, which is simpler to generate than azimuth. Figure 14 shows the average light penetration through a stand with 70% light penetration per cluster and a 2.88x2.88 m (1205 trees per hectare) spacing. The variance is low and constant for all angles. For this tree spacing and biomass level, the curve can be used directly to generate light penetration patterns over time. The output of the cluster level model, in this case, collapses to a single curve that can be used in simple light penetration models as a replacement for exponential decline curves.

Over the course of a year, needle growth and abscission approximately doubles and halves the foliage in the canopy, respectively. In the model there are three ways to simulate the doubling and halving of foliage. The first is to keep the same foliage per cluster and double the number of clusters. This will double the projected cluster area and

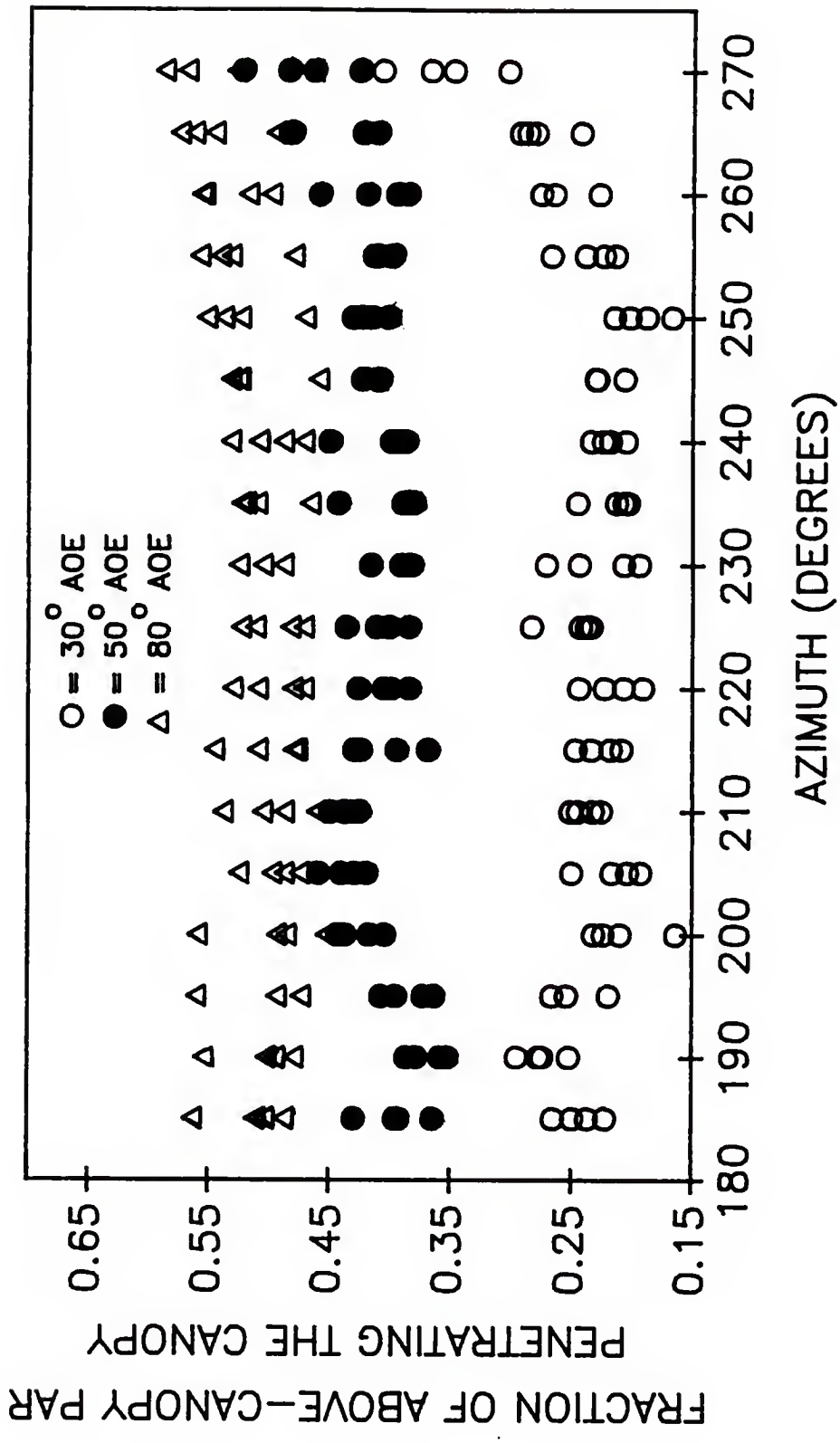

Figure 13: Simulated light penetration through the canopy as a function of AOE and azimuth. Each point represents the average of 196 target locations. Light penetration per cluster was 70%, tree spacing was 2.88x2.88 meters.

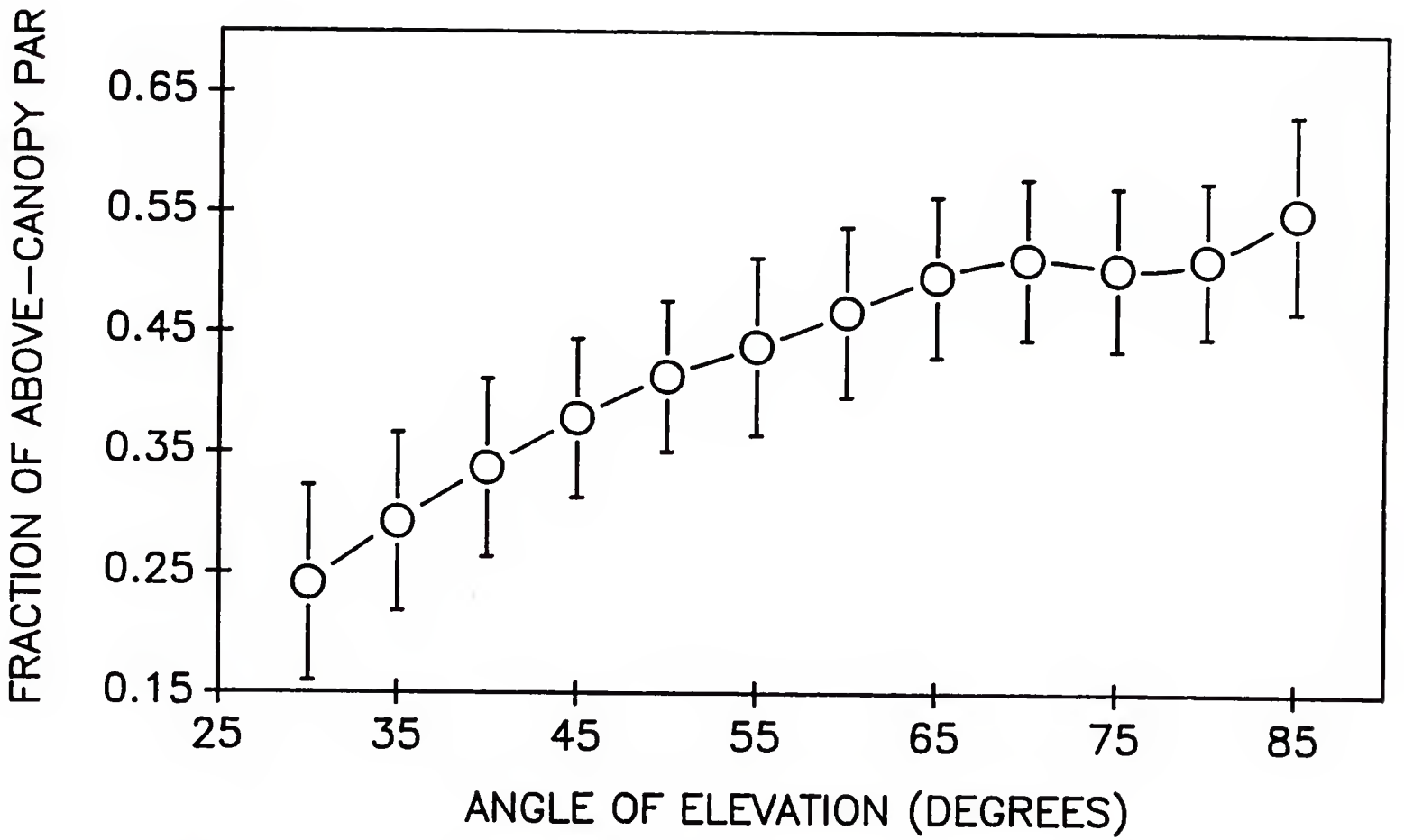

Figure 14: Simulated below canopy light penetration (± 2 SD) as a function of AOE. All azimuths are included. Each point represents 48 simulations.

is roughly equivalent to dominance by the side view of the clusters. The second is to increase or decrease the penetration per cluster according to the developed regression (Figure 11). This is equivalent to dominance by the end view. A third possibility is to increase or decrease the density of the trees. This is not a real possibility within the stand, but halving the number of trees is the approximate equivalent to a thinning operation, leaving wide holes in the foliage between the individual crowns. Doubling the number of trees effectively eliminates between-tree gaps and eliminates much of the conic form of the canopy due to crown overlap. Together these three model runs tested the importance of the cluster and crown levels of hierarchy on light penetration through the canopy.

When these three possibilities were simulated, there was little difference between light penetration patterns generated by the shifting cluster area and shifting cluster density (Figure 15). The differences that did exist occur at high AOE. This suggests that, for below-canopy light penetration, the clusters were acting as if they were randomly located in space. To a large extent they were, because the position of each cluster was randomly chosen within the foliated region of the canopy. This is an

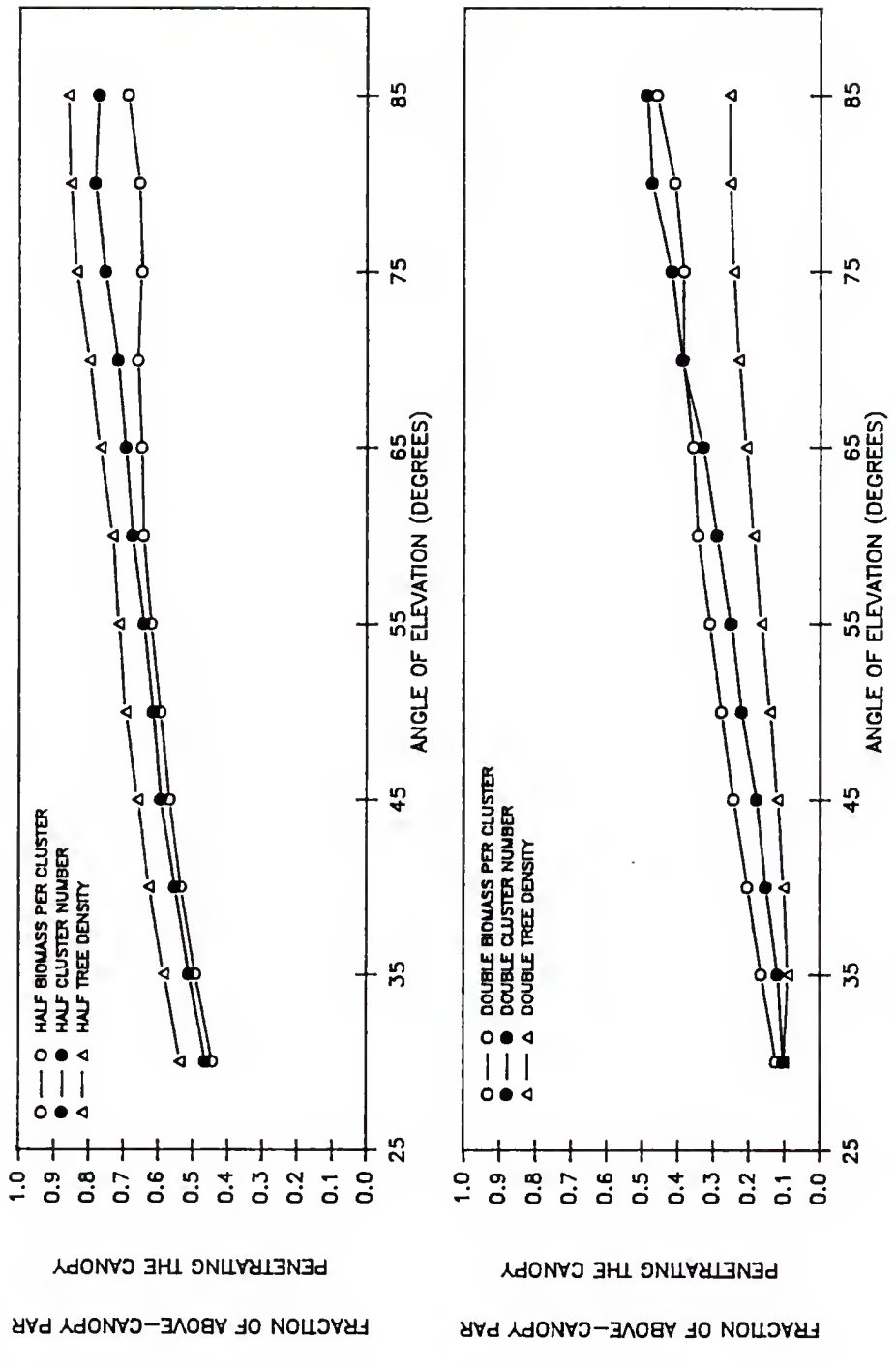

Figure 15: Light penetration through the canopy with changing canopy structure.
A.) Foliage halved.
B.) Foliage doubled.

example of a Beer's law solution in which the parameters are based on cluster rather than needle characteristics.

There were, however, significant differences between light penetration patterns due to changes of biomass within the crown and changes due to changing the number of crowns (Figure 15). These data suggest that the position of the trees is much more important to overall light penetration than is the internal structure of the crowns.

Because the model is insensitive to whether foliage enters the system as additional area or as increased density within the same area, changing biomass within the canopy was modeled by keeping the number of clusters constant and changing their light penetration properties. If the end view projection of the clusters dominates over the side view, or if the clusters are small, the cluster projection area will not change with shifts in biomass. If this is the case, then keeping cluster area constant is appropriate.

Using the cluster level regression (Figure 11), light penetration curves can be created for all sun angles and potential biomass quantities simply by shifting the light penetration per cluster across the range of potential values. This produces a family of curves (Figure 16) representing a specific tree configuration. This family can, in turn, be collapsed to a single equation using

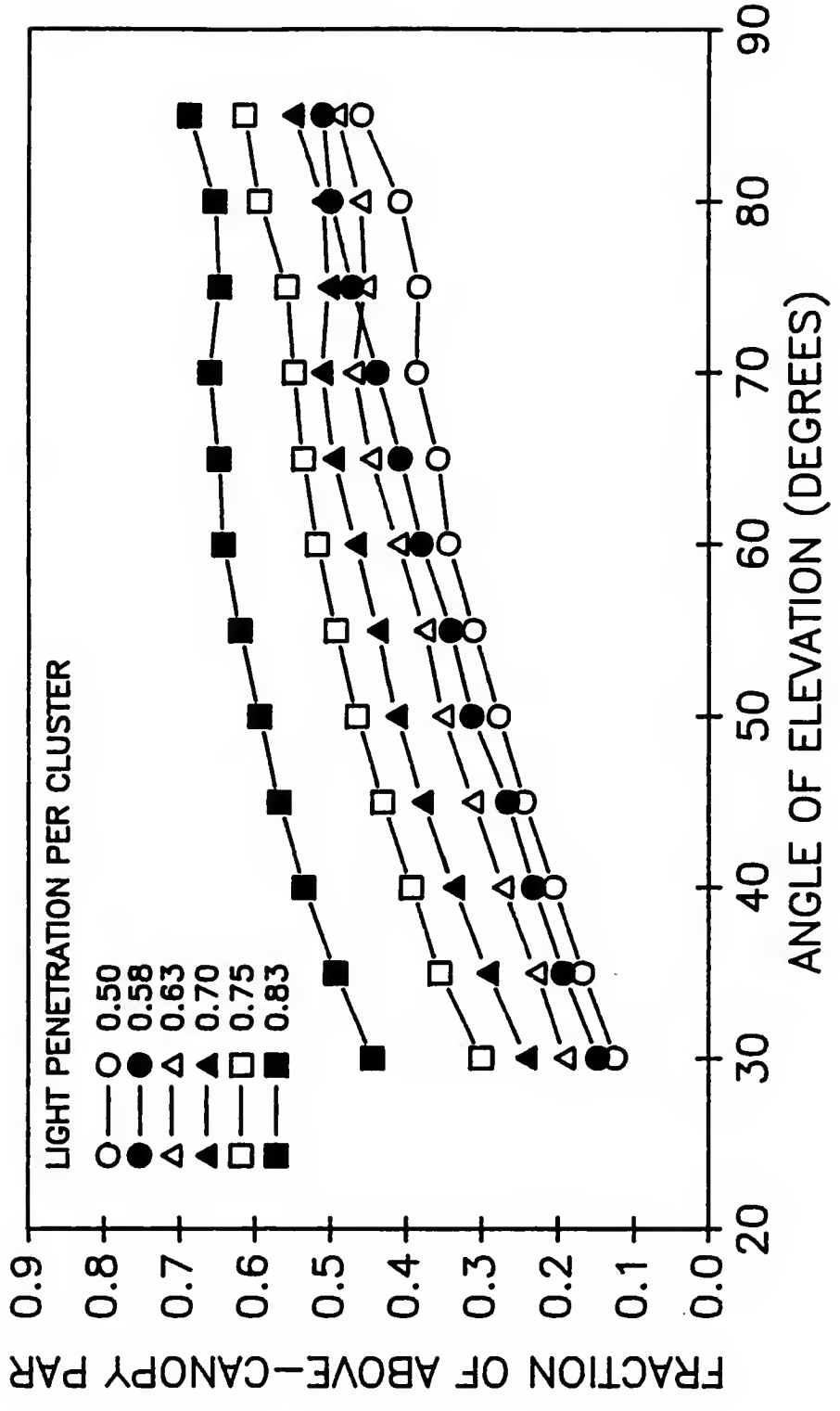

Figure 16: Light penetration through the canopy with constant cluster area and changing biomass. Tree spacing was 2.88x2.88 m.

multiple regression. The equation fit to model output at 2.88 m spacing is

$$I_b = 0.134 + 0.130*AOE - 0.1834*B_f - 6.6E\text{-}5*AOE^2 \qquad (20)$$

$$r^2 = 0.92$$

where

I_b = the fraction of light penetrating to the base of the canopy, and

B_f = a scalar varying between 2.0 and 0.5 with 1.0 producing an average cluster light penetration of 0.7, the approximate median value for the cluster sample.

A cluster level model was constructed for slash pine. The crown form was determined through destructive sampling and the light penetration through individual clusters was measured under controlled conditions. Model output showed that azimuth was not an important parameter for controlling light penetration through the canopy. Most of the model behavior was captured in a regression equation based on biomass and AOE.

MODELING SEASONAL BELOW-CANOPY LIGHT PENETRATION

It is clear from equation 20 that most of the dynamics of the three-dimensional cluster level model can be collapsed to a one-dimensional form. This allows light penetration patterns over time to be modeled using a much simpler model. Other than using equation 20 rather than equation 2 to predict light penetration, a model identical to a Beer's law model for a horizontally homogeneous canopy can be used. Correct AOE can be generated through relatively simple equations, biomass quantities can be entered, and the model can be validated by matching model output with measured light penetration data through the canopy.

Yearly biomass dynamics were derived from three sources of data: destructive sampling, litterfall, and needle elongation rates. Foliage in the canopy at any time is a mixture of the current year's cohort (new foliage) and the previous year's cohort (old foliage). Over a 2-year period, the new foliage becomes old and falls to the forest floor as litterfall. For three consecutive years trees were destructively sampled in August to obtain biomass estimates

for both cohorts. In addition to the biomass estimates described earlier, litterfall was collected in litter traps at bi-weekly intervals (weekly during the peak litterfall period in November). Needle growth was also measured for 15 needles at bi-weekly intervals throughout the season of active growth (weekly during spring bud-break).

Of these data, the destructive sampling was considered to be the most accurate and was used as a baseline for biomass calculation. From these data biomass quantities were calculated both forward and backward through time. The old foliage dynamics were tracked through litterfall. Litterfall was assumed to be entirely old foliage new foliage that fell into the traps due to storm damage was separated. New foliage dynamics were based on needle elongation patterns. Biomass was then converted into all-sided leaf area using a geometric technique (Gholz et al. 1990).

Based on this analysis, it was found that the maximum foliar biomass was approximately 1.7 times the minimum (Figure 17), and that the biomass pattern lagged the solar cycle by approximately three months (Figure 18). In the model, the smoothed biomass curve shown in Figure 18 was scaled to a mean of 1.0 and used as input for equation 20 and other similar equations that represent light penetration through different tree configurations.

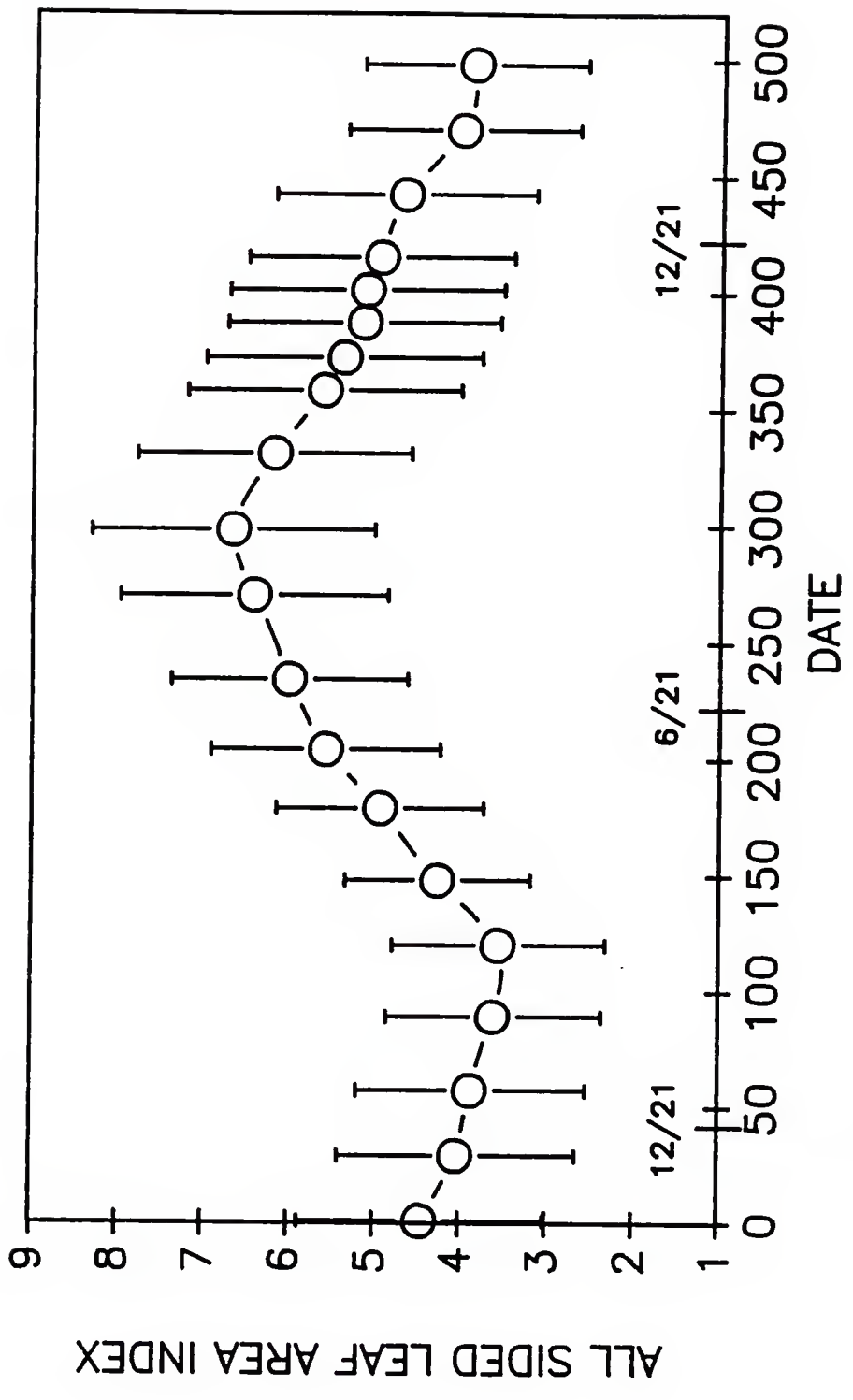

Figure 17: Calculated LAI ($\pm$ 2 SD) over time. The time period starts on 11/7/87 and extends through 3/27/89. Henry Gholz, personal communication.

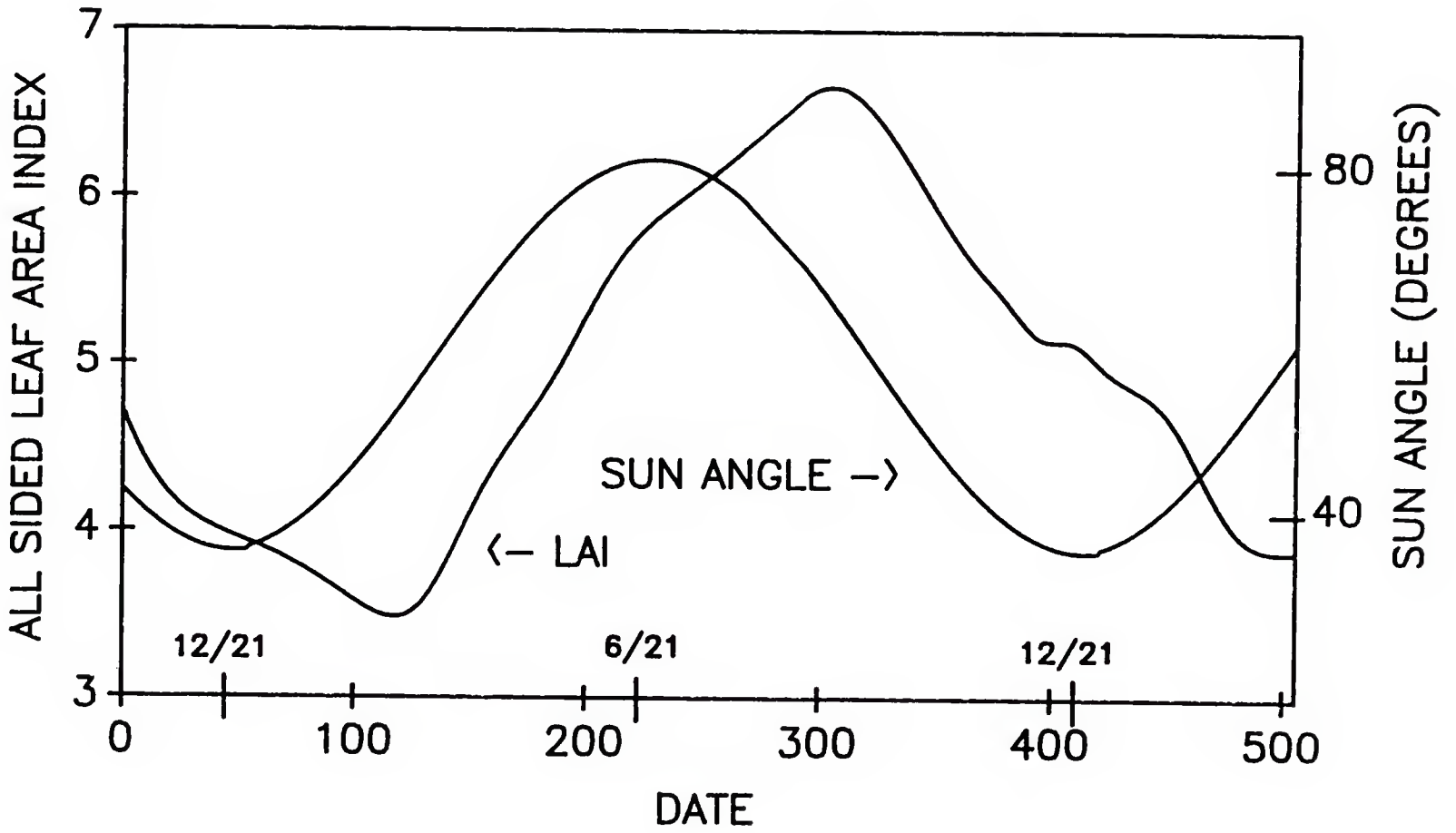

Figure 18: Average LAI and maximum daily sun angle over time.

Light penetration through the canopy was measured with a series of fixed sensor arrays. The arrays consisted of twelve linear response diodes mounted on aluminum beams at 1 m intervals. Each array consisted of two beams placed at right angles on narrow triangular antenna towers. The arrays were level with the ground. Two arrays were placed above the canopy, five were placed immediately beneath the canopy and six were placed above the shrub layer on the forest floor. Readings at each sensor were taken at 5-minute intervals throughout the day and an averaged value recorded was each hour. The sensors were calibrated to Li-Cor PAR spot sensors. Three Li-Cor PAR spot sensors were also placed above the canopy along with two Li-Cor full spectrum sensors. The system was checked for drift over time by regressing hourly above-canopy readings from the PAR spot sensors against the above-canopy diode arrays. An additional check was made by regressing the PAR spot sensors against the full spectrum sensors. The regression between the PAR spot sensors and the above-canopy diode arrays for the period starting on November 6, 1987, through January 8, 1989, was

$$I_1 = -0.508 + 0.9286\ I_s \qquad (21)$$

$$r^2 = 0.99$$

where

I_l = the mean of the two above-canopy diode arrays, and

I_s = the mean of the above-canopy spot sensors.

There is no indication of nonlinear behavior at any light intensity level or any sign of sensor drift.

Because the below-canopy sensors were fixed in space, the possibility exists that the readings were biased by their specific locations. A check was therefore made by regressing the below-canopy sensors against each other in all combinations on an hourly basis (Table 2). These regressions demonstrate that the sensors, both below the canopy and above the shrub layer tracked each other very well (r^2 values ranged from 0.93 to 0.75). Only three of the 36 combinations had an r^2 value of less than 0.8. The slopes, ideally 1.0, ranged from 0.76 to 1.31 with 16 falling below 1.0 and 20 above. The average slope was 1.01. The intercepts, ideally 0, ranged from 47 to -7 (µmol PAR/m2*s). No significant difference was observed between the shrub level sensors and the below-canopy sensors, suggesting that distance from the canopy was not an important influence on light readings.

For determining below-canopy light, all the below-canopy sensors were combined to create an average reading and summed to obtain daily measurements. These daily

Table 2

Regression constants for hourly readings of pairs of below-canopy sensor arrays from November 6, 1987,to October 10, 1989. BC sensors were immediately beneath the canopy, S sensors were above the shrub layer.

Model	r^2	Slope	± SE	Intercept	± SE
		($\mu mol/m^2*sec$)		($\mu mol/m^2*sec$)	
BC1 = BC2	0.90	0.81	0.005	14.69	1.79
BC1 = BC31	0.86	0.97	0.006	24.74	1.61
BC1 = BC32	0.85	0.86	0.005	9.10	1.60
BC1 = BC4	0.88	1.01	0.006	8.69	1.40
BC1 = S1	0.88	1.16	0.006	7.68	1.36
BC1 = S2	0.82	0.86	0.006	5.50	1.80
BC1 = S31	0.89	1.01	0.005	22.00	1.35
BC1 = S32	0.86	0.96	0.006	24.16	1.53
BC1 = S4	0.85	1.17	0.008	23.75	1.50
BC2 = BC31	0.88	1.19	0.008	19.62	2.35
BC2 = BC32	0.88	0.99	0.007	8.65	2.36
BC2 = BC4	0.76	0.97	0.014	32.01	2.93
BC2 = S1	0.82	1.31	0.011	13.49	2.85
BC2 = S2	0.89	1.05	0.006	-0.02	2.02
BC2 = S31	0.93	1.22	0.006	17.90	1.75
BC2 = S32	0.90	1.18	0.007	24.24	2.05
BC2 = S4	0.75	1.08	0.017	42.35	2.90
BC31 = BC32	0.89	0.80	0.004	-3.18	1.25
BC31 = BC4	0.81	0.93	0.008	5.01	1.68
BC31 = S1	0.81	1.05	0.008	5.66	1.85
BC31 = S2	0.85	0.80	0.005	-6.71	1.56
BC31 = S31	0.92	0.97	0.004	4.53	0.99
BC31 = S32	0.89	0.91	0.005	9.13	1.20
BC31 = S4	0.77	1.05	0.011	19.28	1.81
BC32 = BC4	0.82	1.02	0.008	31.61	1.82
BC32 = S1	0.83	1.21	0.008	24.17	1.84
BC32 = S2	0.89	0.96	0.005	6.51	1.50
BC32 = S31	0.92	1.12	0.004	26.54	1.11
BC32 = S32	0.90	1.07	0.005	27.07	1.25
BC32 = S4	0.80	1.18	0.009	47.54	1.86
BC4 = S1	0.86	1.06	0.007	6.92	1.44
BC4 = S2	0.81	0.76	0.006	3.78	1.69
BC4 = S31	0.85	0.93	0.006	17.72	1.47
BC4 = S32	0.83	0.85	0.006	20.12	1.54
BC4 = S4	0.84	1.09	0.007	24.20	1.36

figures were then divided into the daily sum of the average of the two above-canopy sensor arrays to obtain fractional below-canopy light penetration daily beginning November 7, 1987, and continuing through March 27, 1989. A period in August 1988 is missing due to a lightning strike and several other days are also missing due to equipment failure. Light estimates were made for 473 days of a potential 505 days.

With good data sets for both below-canopy light penetration and biomass fluctuation, light penetration over time can be modeled and validated. To obtain accurate estimates of daily light integrals, a simple model can be used to generate sun angles and approximate intensities (Campbell 1977). For sine-corrected light impacting on a flat surface,

$$\text{SinAOE} = \sin((\sin(L)\sin(\text{Dec}) + \cos(L)\cos(\text{Dec})\cos[15(t-t_0)]) \quad (22)$$

where

SinAOE = the sine of the AOE,

t = the time of day in hours (24 hour clock),

t_0 = solar noon,

L = the latitude in degrees, and

Dec = the declination in degrees.

Sine corrected solar intensity on flat ground at sea level can be approximated by

$$I_f = \sin[I_{sc} * T_a/\text{SinAOE}] \quad (23)$$

where

I_f = the solar intensity on flat ground,

I_{sc} = the solar constant, and

T_a = atmospheric transmissivity.

For determining proportional light penetration, I_{sc} can be set to 1.0, and is dimensionless. T_a is also dimensionless and varies from 0.9 on very clear days to 0.6 on hazy or polluted days (Campbell 1977).

Light penetration through the canopy was calculated at hourly intervals using equations 20, 22, and 23. To obtain daily time integrals, the results were assumed to be constant for the duration of the hour. This integration technique is justified for daily time integrals because, due to the symmetry in angular distribution around solar noon, integrated light will be underestimated before noon and overestimated after noon by approximately equal amounts.

The time series extended from November 7, 1987, to March 27, 1989, the period when above and below-canopy sensors were in place and biomass estimates were computed.

If, for a tree spacing of 2.88x2.88 m, light penetration per cluster is assumed to be 70% throughout the time series, the simulation matches the data until the summer of 1988 and then diverges (Figure 19). Light penetration is overestimated for the remainder of the time series. If, however, it is assumed that 70% light penetration per cluster represents the minimum biomass levels (March 1988), and biomass is allowed to fluctuate over time according to the pattern shown in Figure 18, the fit is greatly improved (Figure 19).

Diffuse light penetration of the canopy differs from direct beam radiation because it emanates directly from all portions of the sky dome simultaneously. The quantity of diffuse light reaching a target point within the canopy will be decreased by an amount proportional to the area of the sky dome that is obscured by surrounding foliage and the light intensity of those portions of the sky that are obscured. The obscured portion of the sky dome, in turn, is related to the distance that the shade object is from the target and the size of the shade object. One approach to diffuse modeling is therefore to compute sky dome coverage for each target point. This is the approach used by Norman and Welles (1983) for envelope modeling.

The chief problem with this approach lies in the determination of the actual patterns of diffuse radiation.

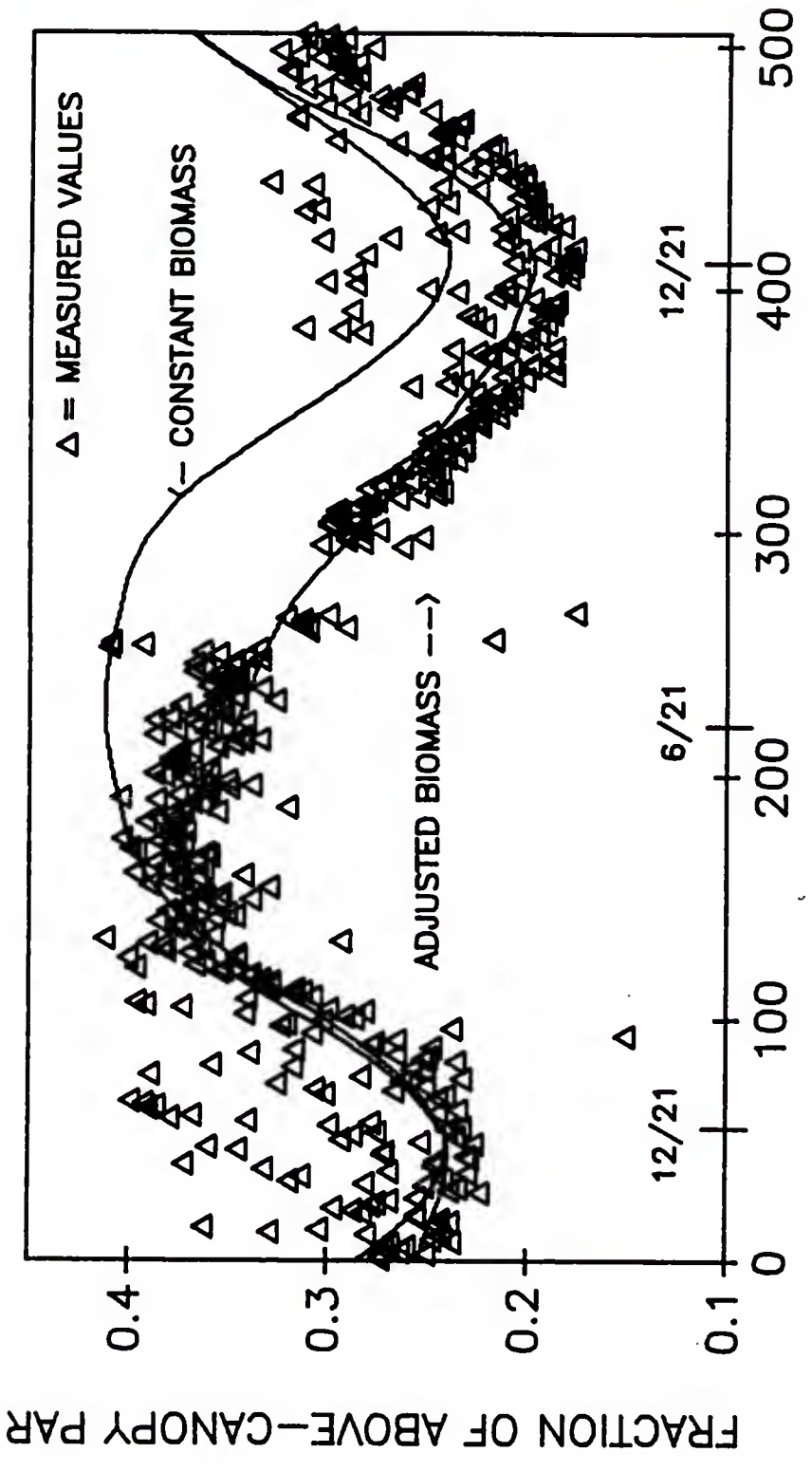

Figure 19: Simulated light penetration through the canopy compared with daily data. Initial cluster penetration was 70%, tree spacing was 2.88x2.88 m. Changes in biomass followed the LAI pattern in Figure 18.

Diffuse patterns are quite predictable when the sky is heavily overcast (Figure 20) but become very complex on sunny days (Grant 1985). Diffuse light patterns are therefore difficult to simulate because they shift with the weather and cannot be generated through trigonometric techniques. In clear sky conditions, direct light dominates and the pattern of diffuse light is strongly anisotropic. Under heavy overcast, diffuse light is isotropic. On uniformly overcast days, the intensity per unit area of the sky dome is greatest at the top of the dome and least at the horizon. The area, however, is greatest at the horizon. The greatest quantity of light therefore emanates from the mid-sky region (Figure 20). Because azimuth was unimportant to light penetration into the stand, a very simple way of modeling diffuse light is to assume that it all emanates from the mid-sky.

The premise that diffuse light enters the stand primarily from the mid-sky should have the result that, in the winter, days dominated by diffuse light should have higher light penetration through the canopy than clear days. This is because, in North Florida, the maximum AOE for direct beam radiation at the winter solstice is 38 degrees. No direct light radiation enters the canopy from the mid-sky (AOE = 45 degrees). In the summer, the effect of diffuse light on below-canopy light penetration should

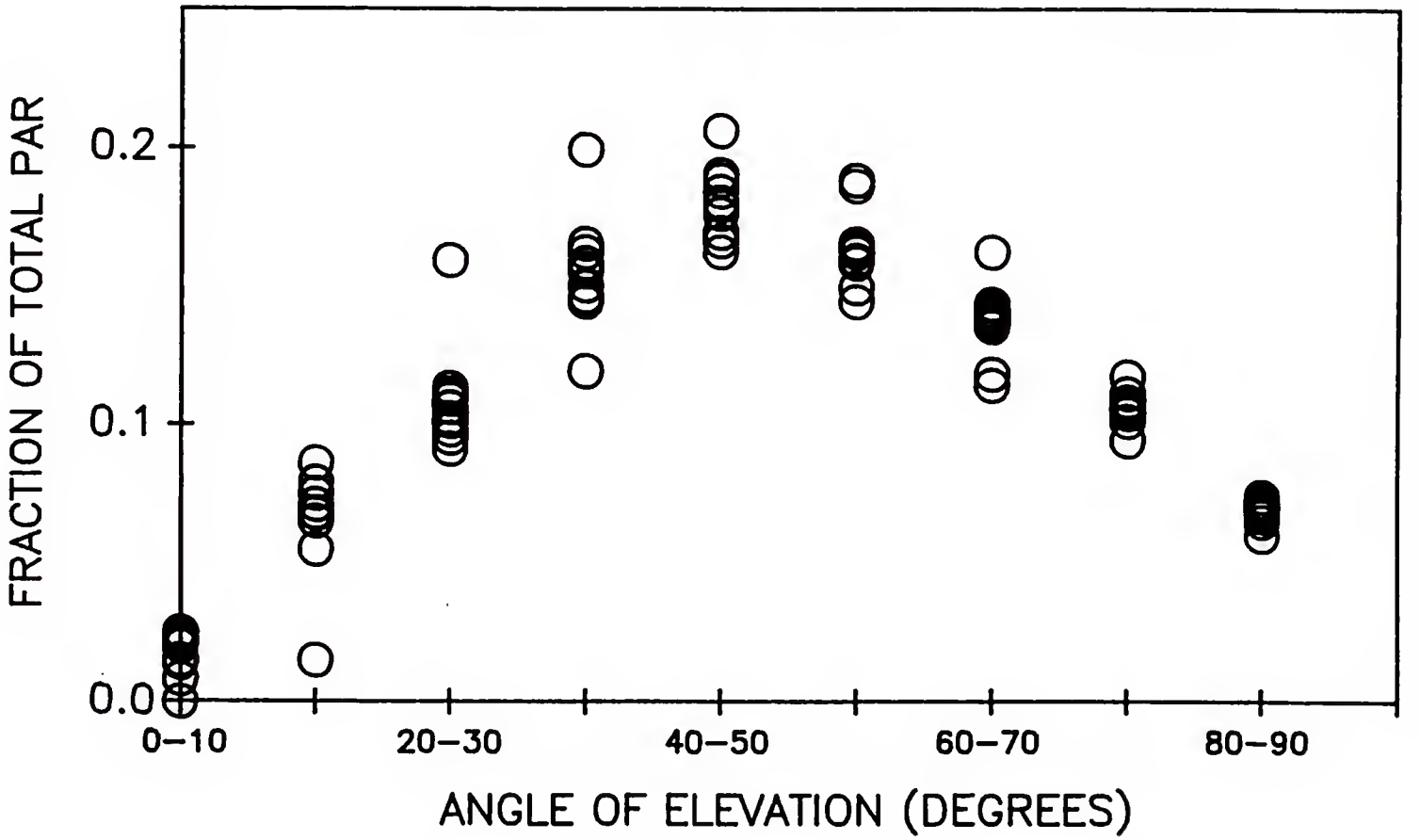

Figure 20: Measured values for diffuse light as a function of AOE. Readings were taken on uniformly overcast days.

be much more muted because a great deal of the direct light emanates from the mid-sky. If there is a pattern, it should be reverse from the winter pattern because mid-day direct light emanates from AOE as high as 84 degrees.

For a given sun angle, changes in above-canopy PAR are due to differences in atmospheric transmissivity. Decreased transmissivity, due to water or dust, increases the proportion of diffuse light (Campbell, 1977; Spitters et al. 1986). For periods with similar sun angle distributions, such as exist around the solstices, days with the lowest integrated PAR should have the highest proportion of diffuse light and those with the highest PAR should have the lowest. When a two month period surrounding each solstice was plotted against integrated PAR, the patterns supported the hypothesis that diffuse light emanated primarily from the mid-sky (Figure 21). For the period extending one month on either side of the winter solstice, there was a strong negative correlation between high above-canopy PAR and below-canopy light penetration. For the period extending one month to either side of the summer solstice, there was a very weak positive correlation.

Based on this empirical support, diffuse light was added to the direct light model. It was generated by using the following rules:

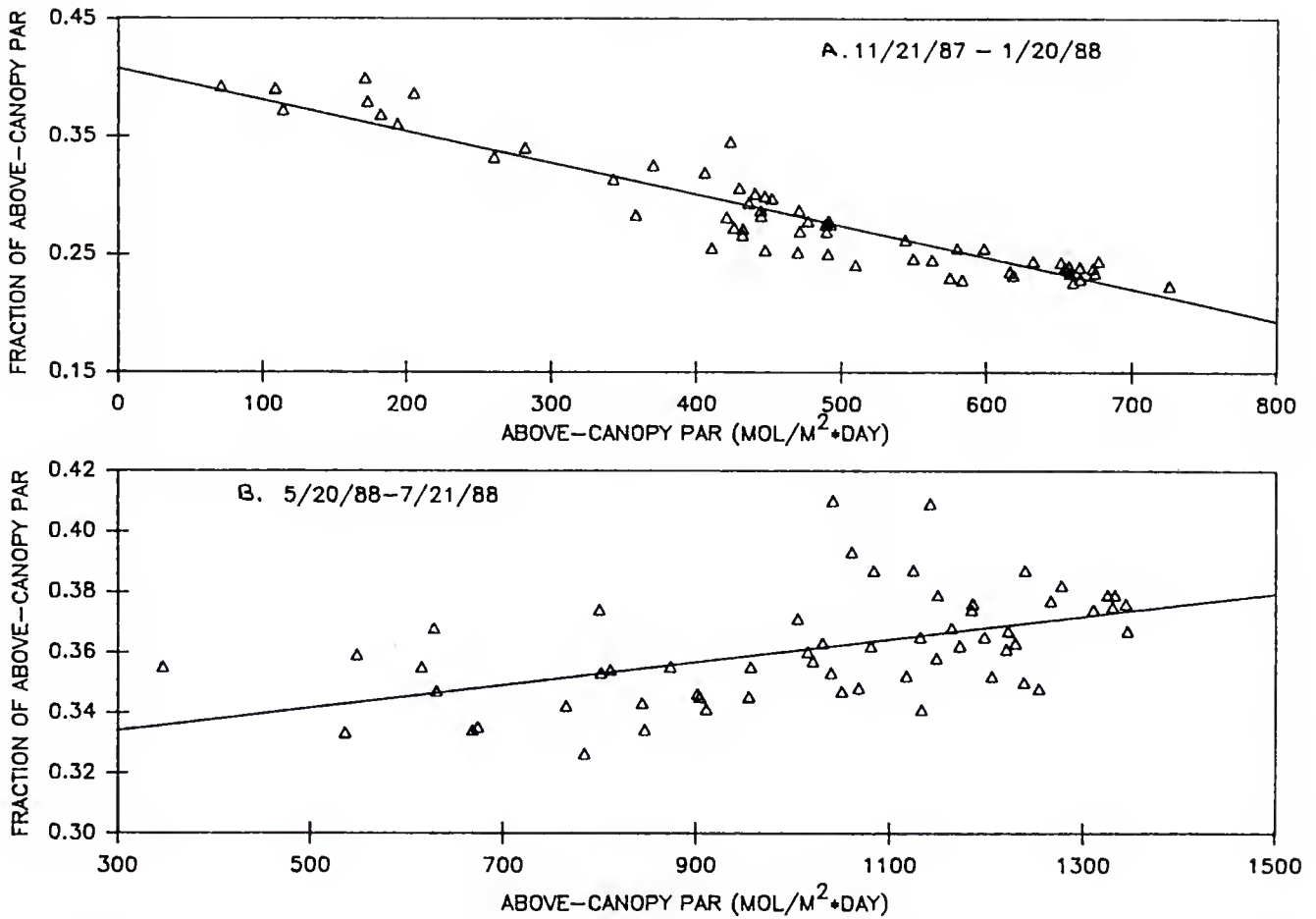

Figure 21: Light penetration through the canopy as a function of total daily PAR
A.) A two-month period centered on the winter solstice.
B.) A two-month period centered on the summer solstice.

1.) 25% of the days, randomly chosen, were not sunny;
2.) the split between direct and diffuse light was purely random on those days (some would be partly cloudy);
3.) all diffuse light emanated from 45 degrees AOE.

When these rules were applied to the annual light penetration model, patterns of winter variance and summer constancy in light penetration were duplicated with the exception of early summer variation (Figure 22). In the early summer, the measured data displayed higher variation than the model predicted. This suggests that this approach to modeling diffuse light captures a great deal of the impact of cloudy weather on canopy below-canopy light penetration, and that the pattern of variation observed in below-canopy light is caused by differences in system sensitivity to diffuse radiation rather than differences in weather pattern.

The rules for splitting days into diffuse and direct light listed above were largely arbitrary. No attempt was made to match light penetration for specific days. The proportion of clear and cloudy days was, however, reasonably accurate. When total daily radiation for 699 days was compared with theoretical clear sky values, 75.4% of the days were within 25% of the theoretical values. The strong correlation between total above-canopy PAR and below-canopy light penetration seen in Figure 21a, however,

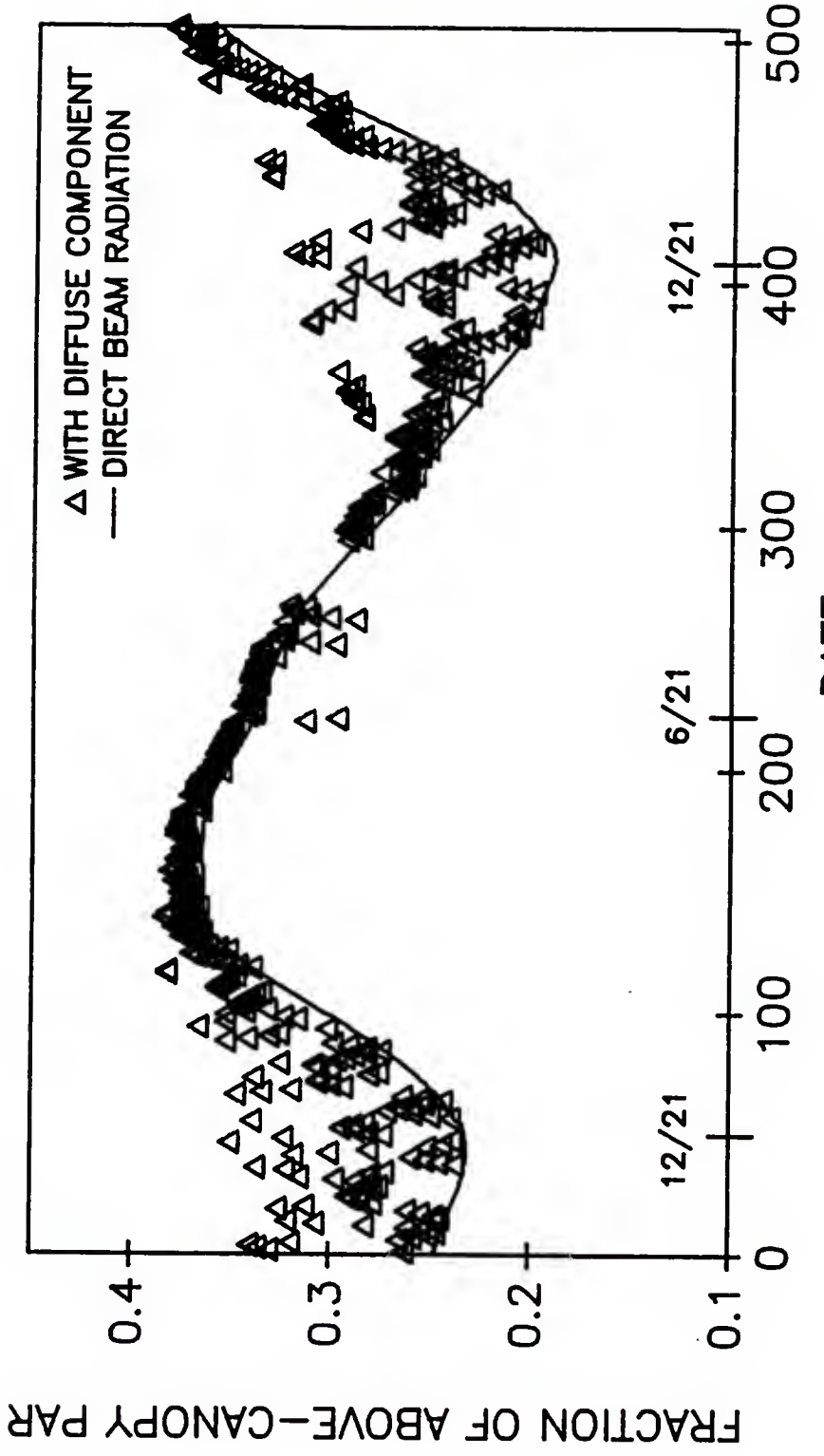

Figure 22: Simulated penetration of light with a diffuse component through the canopy.

suggests that light penetration can be simulated on a daily basis if a rule for partitioning the total light into diffuse and direct components based on daily above-canopy PAR can be derived.

The simplest pattern is to assume that the proportion of diffuse light is a linear function of above-canopy PAR. For the period covered in Figure 21a, it was assumed that daily PAR of 100 mol represented a uniform overcast day with 100% diffuse radiation and PAR of 800 represented a day in which all of the light entered the stand as direct radiation. When this linear pattern is modeled, the simulation tracks the pattern of the measured data, but has lower variation (Figure 23a). This suggests that the shift from direct to diffuse light is a nonlinear function of daily PAR. In its most extreme form, this nonlinearity will take the form of a switch where the light pattern shifts from entirely direct to entirely diffuse at a specific daily PAR.

Campbell (1977) breaks diffuse modeling into a clear sky model, which allocates 10% to 30% of the total radiation to diffuse (depending on the amount of haze) and a cloudy sky model in which all of the light is assumed to be diffuse. In Campbell's system, all light is assumed to be diffuse whenever clouds obscure the sun. For a series of daily light integrals, a variety of cloud patterns will

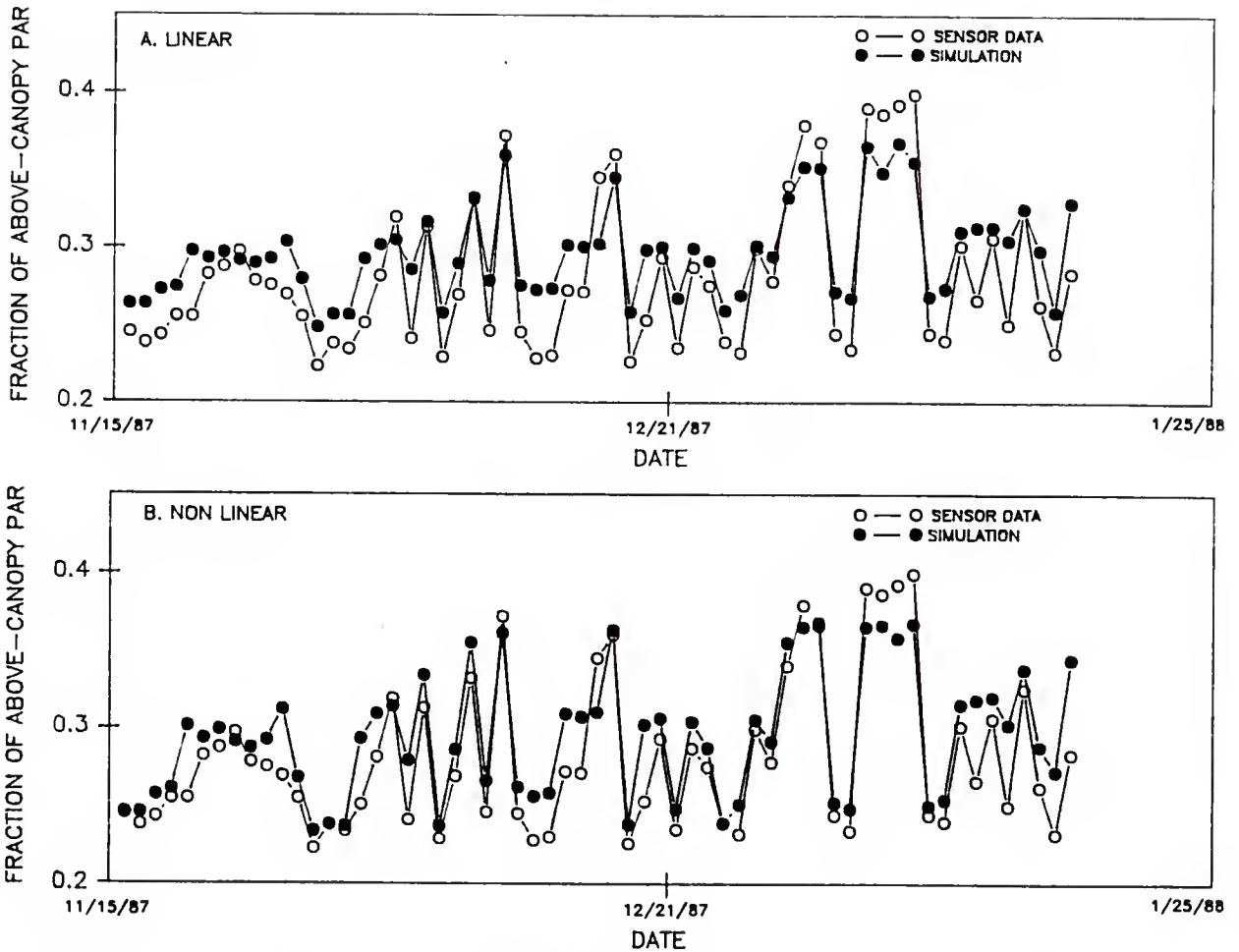

Figure 23: Simulated and measured daily PAR around the winter solstice.
A.) Linear model.
B.) The nonlinear model uses a sigmoid curve to partition light.

exist. If a day has high atmospheric transmissivity and is partly cloudy then, for part of the day, the clear sky conditions will dominate. Under these conditions, it there is a linear relationship between daily integrated light and the percent cloud cover (Spitters et al., 1986). Campbell's models suggest that the proportion of diffuse radiation will also be linearly dependent on the percent cloud cover. Frontal weather will produce either uniformly cloudy or uniformly clear days, and the dynamics should be much more like a switch. A sigmoid is a simple nonlinear function with dynamics intermediate between a line and a switch, and this pattern has been shown to fit empirical data for many sites (Spitters et al. 1986). When a sigmoid function replaces the linear dependency, the model fit is improved (Figure 23b). When the measured data are regressed on the simulation, both models account for 83% of the natural variation. The slope, ideally 1.0, is 1.56 for the linear model and 1.07 for the sigmoid model.

Simple models for diffuse radiation, when coupled with the regression obtained through cluster level modeling, duplicate the observed pattern of light penetration through the canopy on both a seasonal and a daily scale.

Because the output from the cluster level model was reduced to a single regression equation, it was possible to contrast the model results with a Beer's law model driven

by identical input. Assuming constant foliage, the only divergence lay at high sun angles (Figure 24). When the biomass dynamics were included, the Beer's law model matched identically in the winter and spring, but was more sensitive to the biomass buildup during the summer (Figure 25). Compared with the measured light data, the Beer's law model provided a poorer fit to the summer light penetration than did the cluster level model (Figure 26) but, in general, a sine-corrected Beer's model fits the light patterns reasonably well.

To determine the extent to which the difference between the Beer's law model and the cluster level model was based on the presence of clustering, a modified Beer's law model was constructed. In this model, it was assumed that clusters, rather than leaves, were the basic unit. The clusters were randomly positioned, and the light penetration through the clusters was based on the regression derived through direct measurement (Figure 11). When output from this model is compared with output from the discrete cluster model, the patterns are very similar for the 2.88 m spacing (Figure 27). This provided further evidence that, in the cluster-level model, clusters were projecting randomly onto the forest floor. The close fit between these two models also suggested that the discrete

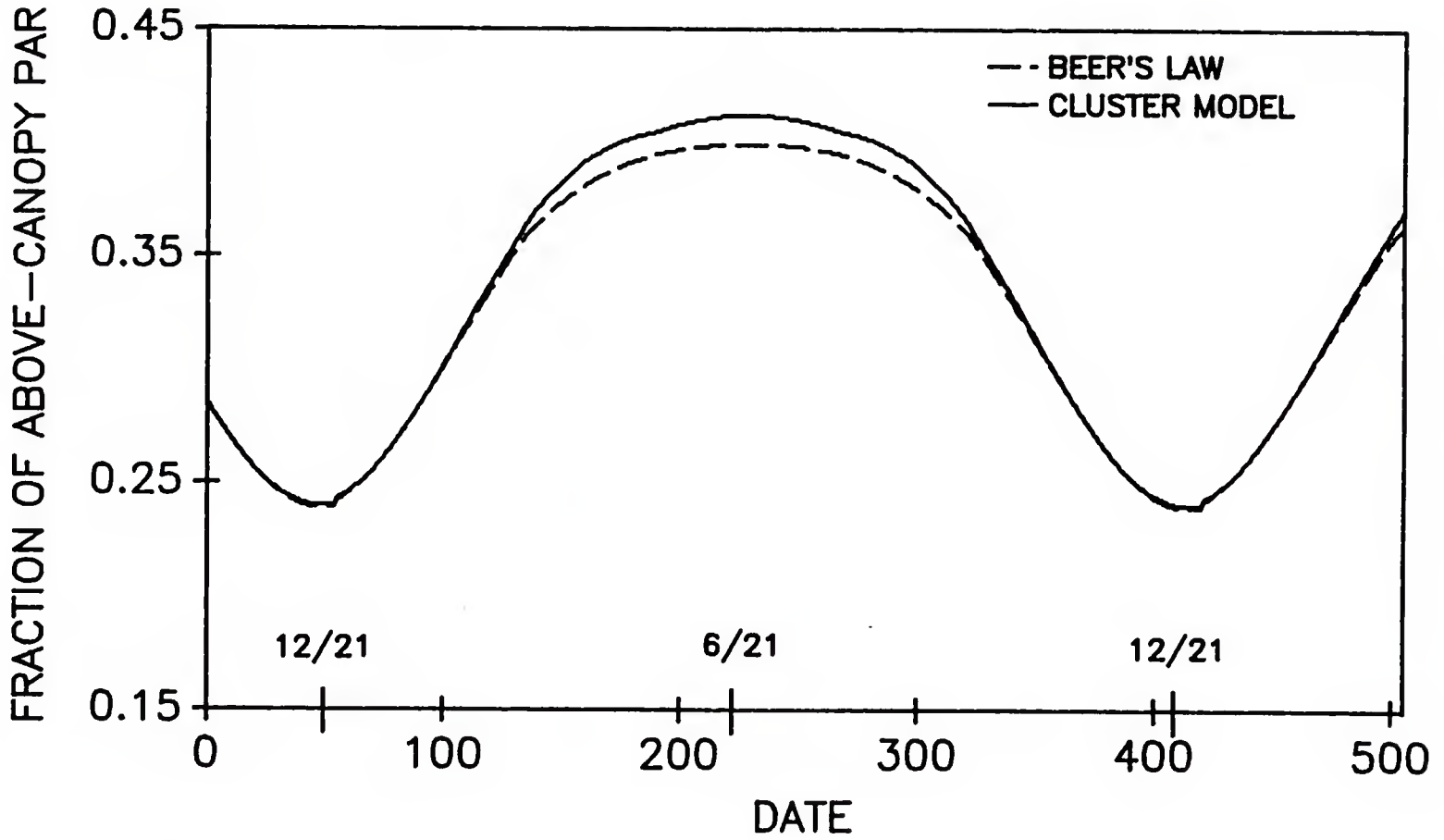

Figure 24: Comparison of a regression model and a Beer's law model with constant biomass.

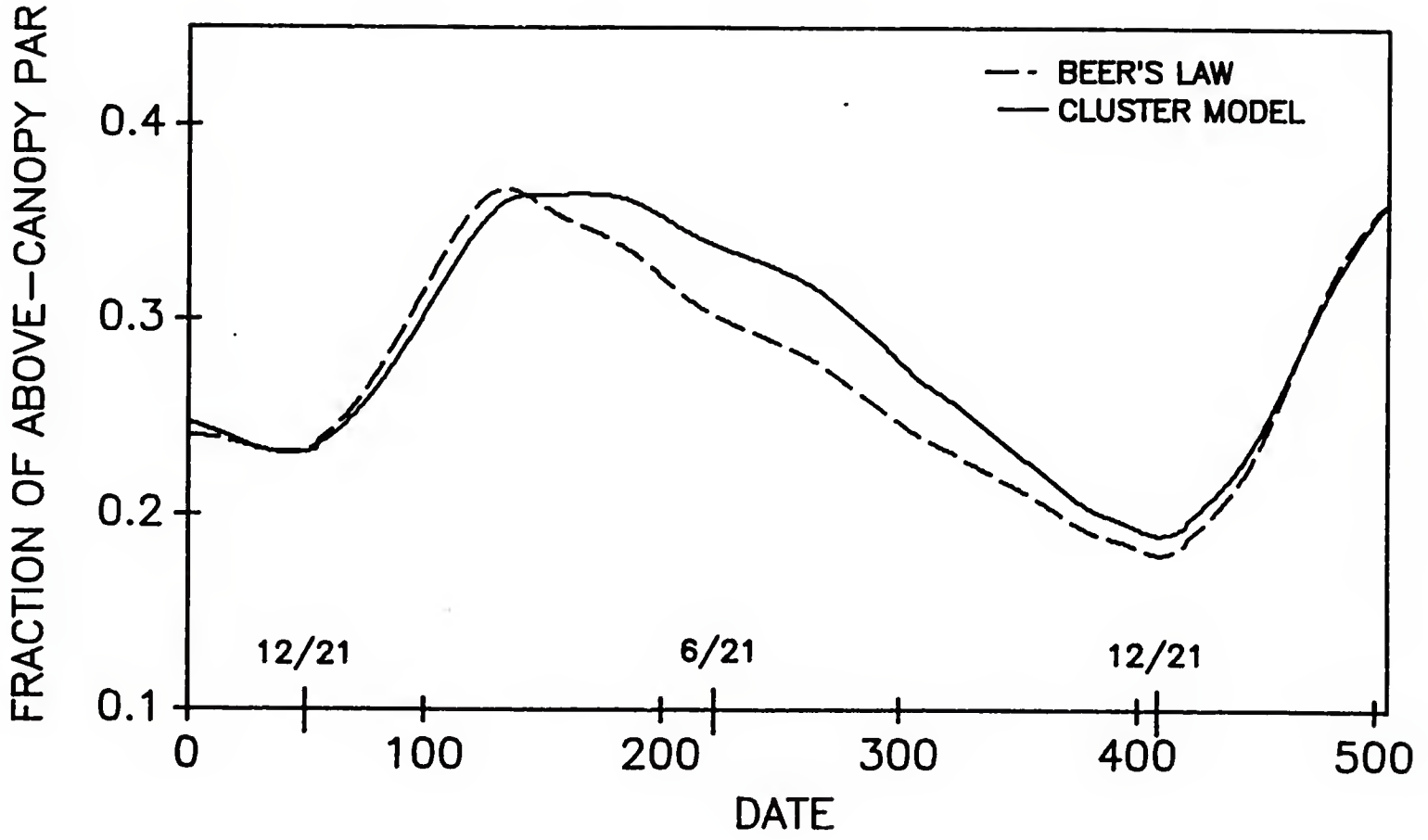

Figure 25: Comparison of a regression model and a Beer's law model with changing biomass. Biomass shifts followed the LAI pattern in Figure 18.

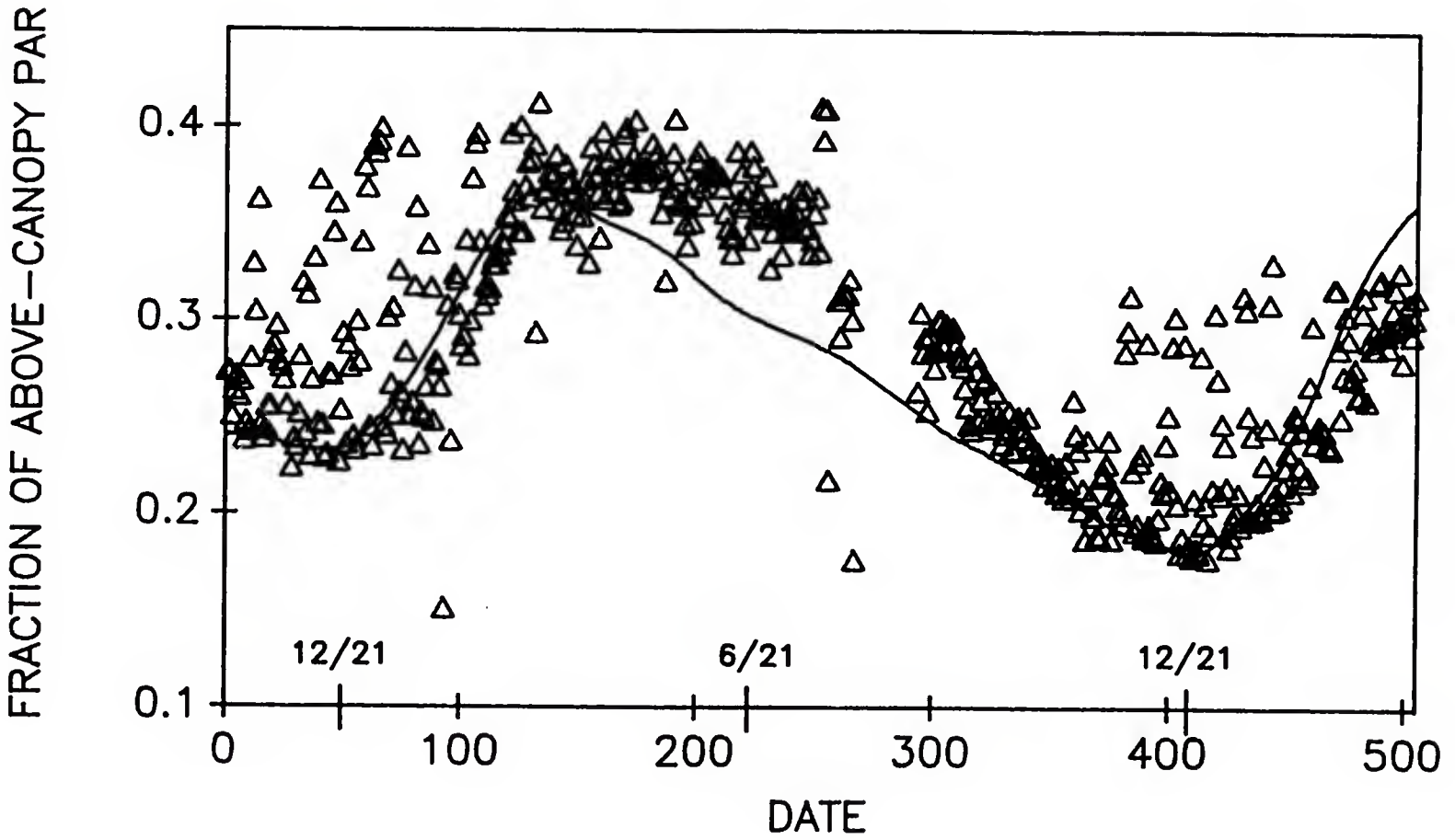

Figure 26: A Beer's law model fit to the below-canopy light data. The model was run using a time step of 1 hour. Biomass followed the pattern of LAI in Figure 18.

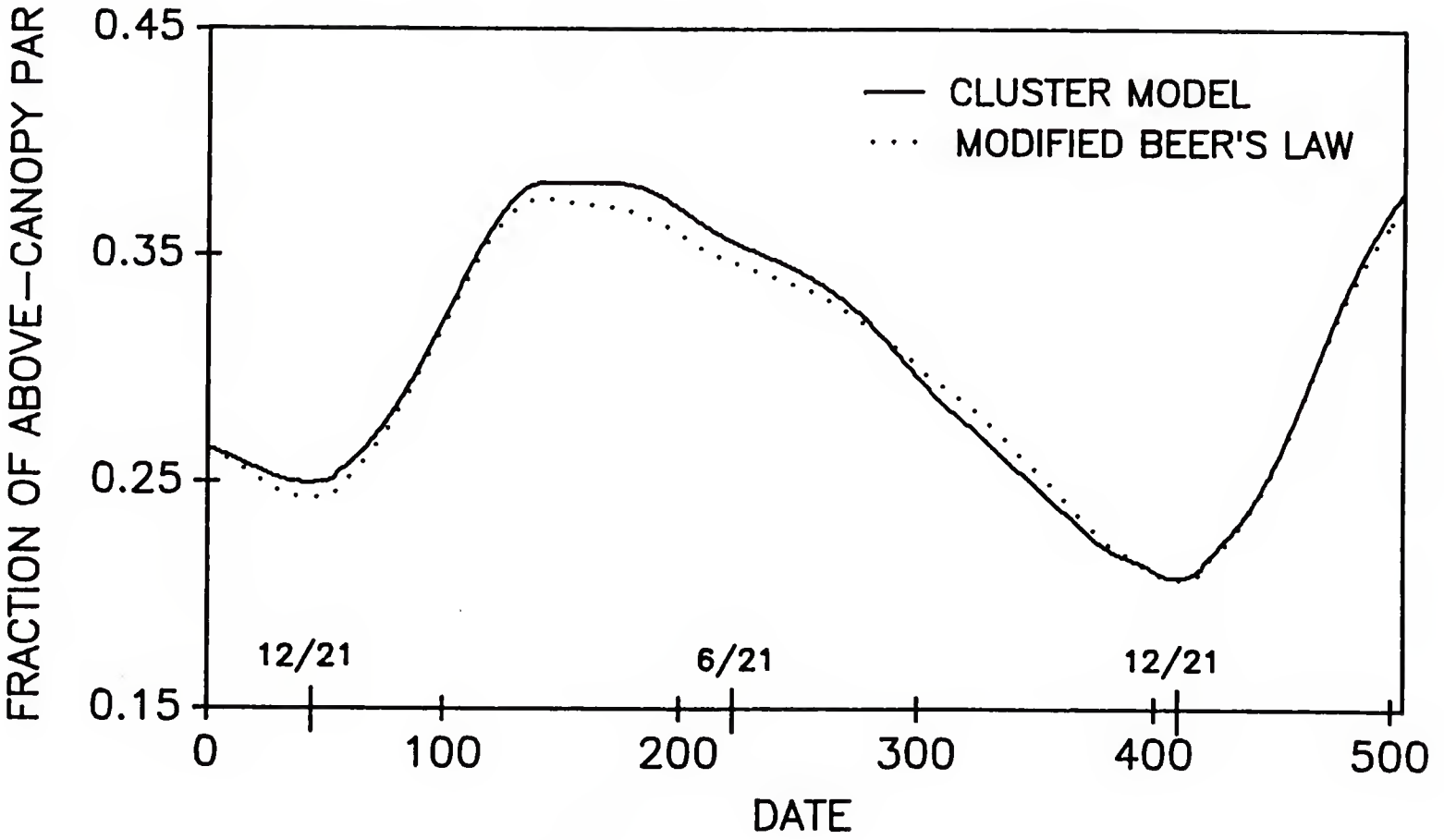

Figure 27: Simulated light penetration using a regression model and a modified (based on clusters rather than leaves) Beer's law model. The model was run using a time step of 1 hour. Biomass followed the pattern of LAI in Figure 18.

representation provided a numerically accurate picture of light penetration.

MODELING LIGHT PENETRATION WITHIN THE CANOPY

Light penetration patterns through the canopy are much more difficult to obtain than below-canopy values but are more important to canopy physiology. The cluster-level model is capable of simulating penetration either to a planar surface within the canopy structure or to a target tree in which light penetration is calculated to the surface of each cluster.

To obtain a rough estimate of light penetration to a series of planes within the canopy, five profiles were run on a uniformly overcast day (Figure 28). An overcast day was chosen to minimize variation within the canopy. The profiles were created by hanging off a small triangular antenna tower on the control plot and swinging a line sensor in a circle approximately 2 m in radius. Light readings were taken as much as possible at 15 degree intervals along the arc. Each arc was therefore characterized by 25-30 individual readings which were then averaged to obtain a single reading representing the arc. Arcs were spaced at 1-m intervals from the approximate top of the canopy (ascertained by sighting with a clinometer)

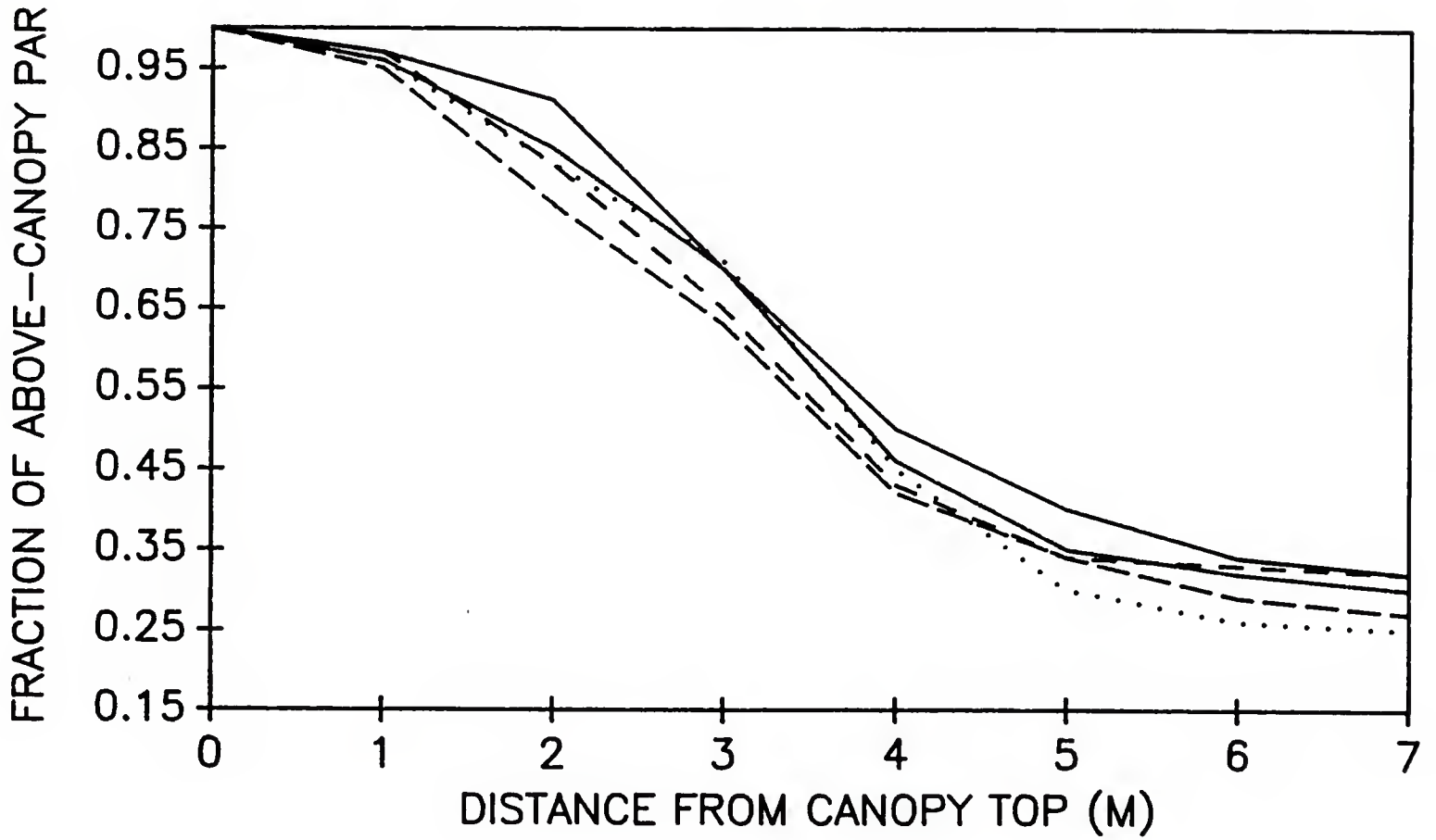

Figure 28: Light penetration profiles through the canopy. Each line represents an individual profile. Readings were taken on a uniformly overcast day.

to a depth of 7 m. The 7-meter reading was entirely below the canopy.

When light profiles to a series of planar surfaces were modeled, the measured values fell between the simulated curves for 30 and 50 degrees AOE (Figure 29). The model therefore fit the data for a mid-sky AOE, as is appropriate for uniform overcast conditions.

It is fair to question the biological meaning of light penetration profiles based on a series of planar surfaces within the canopy. At the top of the canopy, the line sensor was primarily sweeping empty space, measuring the holes between the crowns. At the base of the canopy, it was primarily measuring the non-foliated zone surrounding the stem, an area that presumably does not provide a favorable environment to support foliage. When it was assumed that the planes existed only at spatial locations representing clusters, the curves looked quite different (Figure 30). A strong self-shade depression was indicated for high AOE in the zone immediately below the layer of maximum foliar biomass (Figure 6). In this region, the crown was still narrow (Figure 3) and some self shading was inevitable. The extent to which the simulated self-shade depression was accurate depends on the validity of the rules for cluster placement. In the model, clusters were placed randomly in space. On the trees they are associated with limbs and

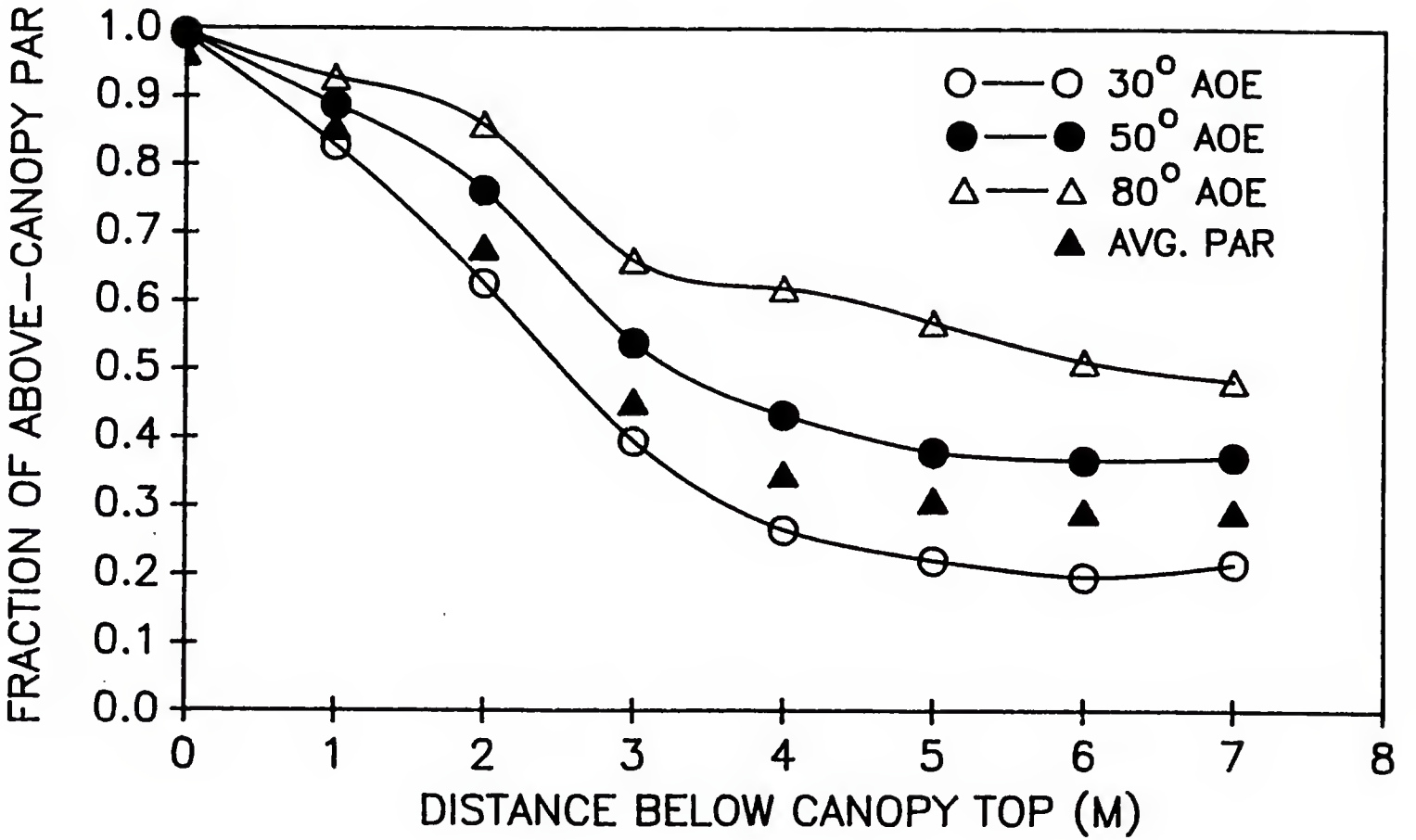

Figure 29: Simulated light penetration profiles through the canopy and average measured data (n=5). Cluster penetration was 70%, tree spacing was 2.88x2.88 m.

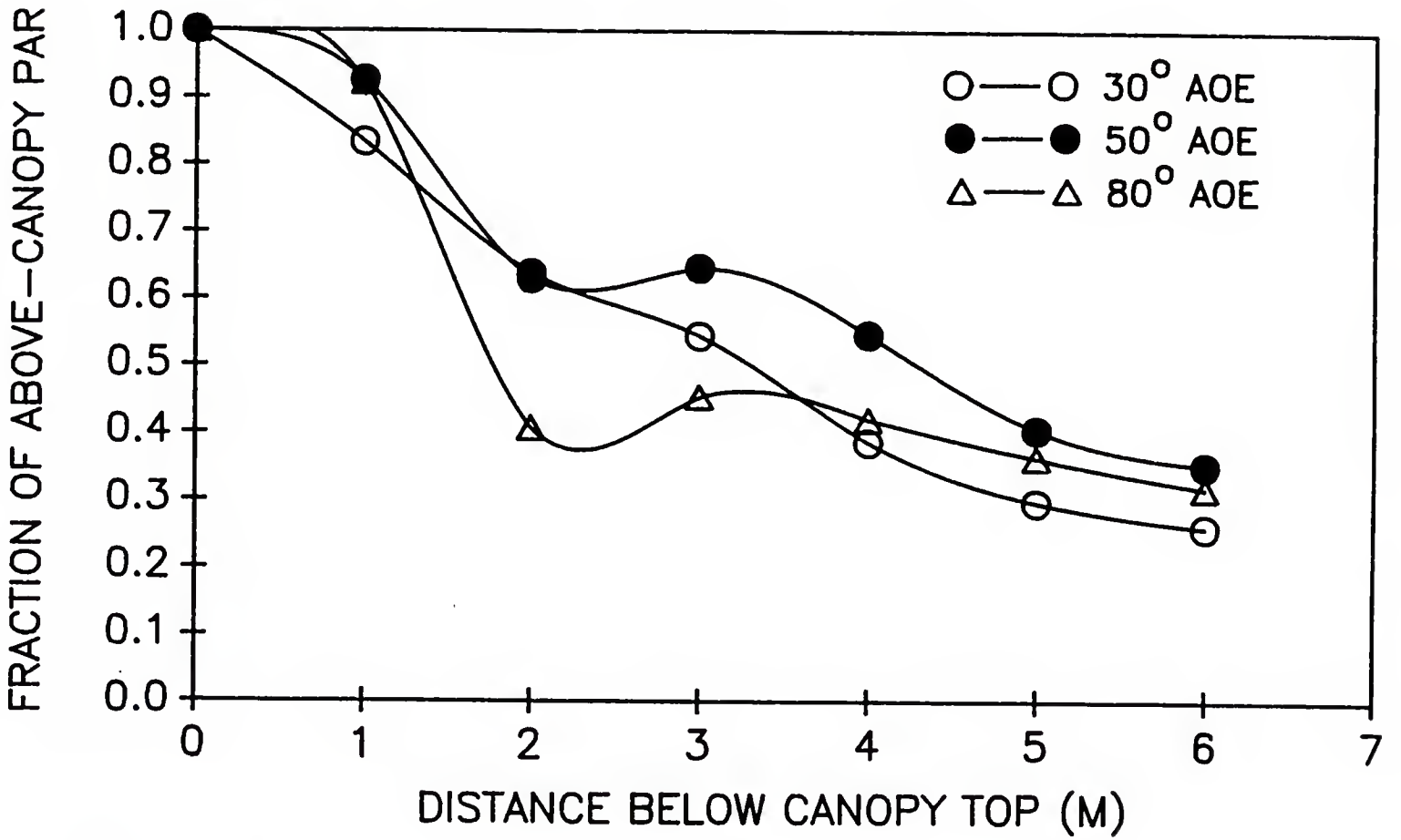

Figure 30: Simulated light penetration to the cluster surfaces. Cluster penetration was 70%, Tree spacing was 2.88x2.88 m.

presumably can grow out of each other's way. A self-shade depression should exist, but it is expected to be smaller than was simulated.

Stratified Beer's law models were constructed for both the biomass profile adjusted for branch angle and the unadjusted profile (Figure 6). When fit to the below-canopy light penetration levels, the model based on the adjusted biomass underestimated the light penetration in the upper canopy, but the model based on the unadjusted profile provided a nearly perfect fit (Figure 31). Because the unadjusted profile is clearly incorrect (It assumed that no foliage existed above the base of the leader.), this fit was largely fortuitous. Returning to the discussion of heterogeneity in the introduction, crowns are small and dense near the top of the canopy. $P_{no\ crown}$ (equation 5) will therefore be large (Figure 1), and a homogeneous model should underestimate the light penetration. The method for computing the unadjusted biomass profile shifted the foliage down from its actual position and served to compensate for the underestimation of light penetration per unit biomass.

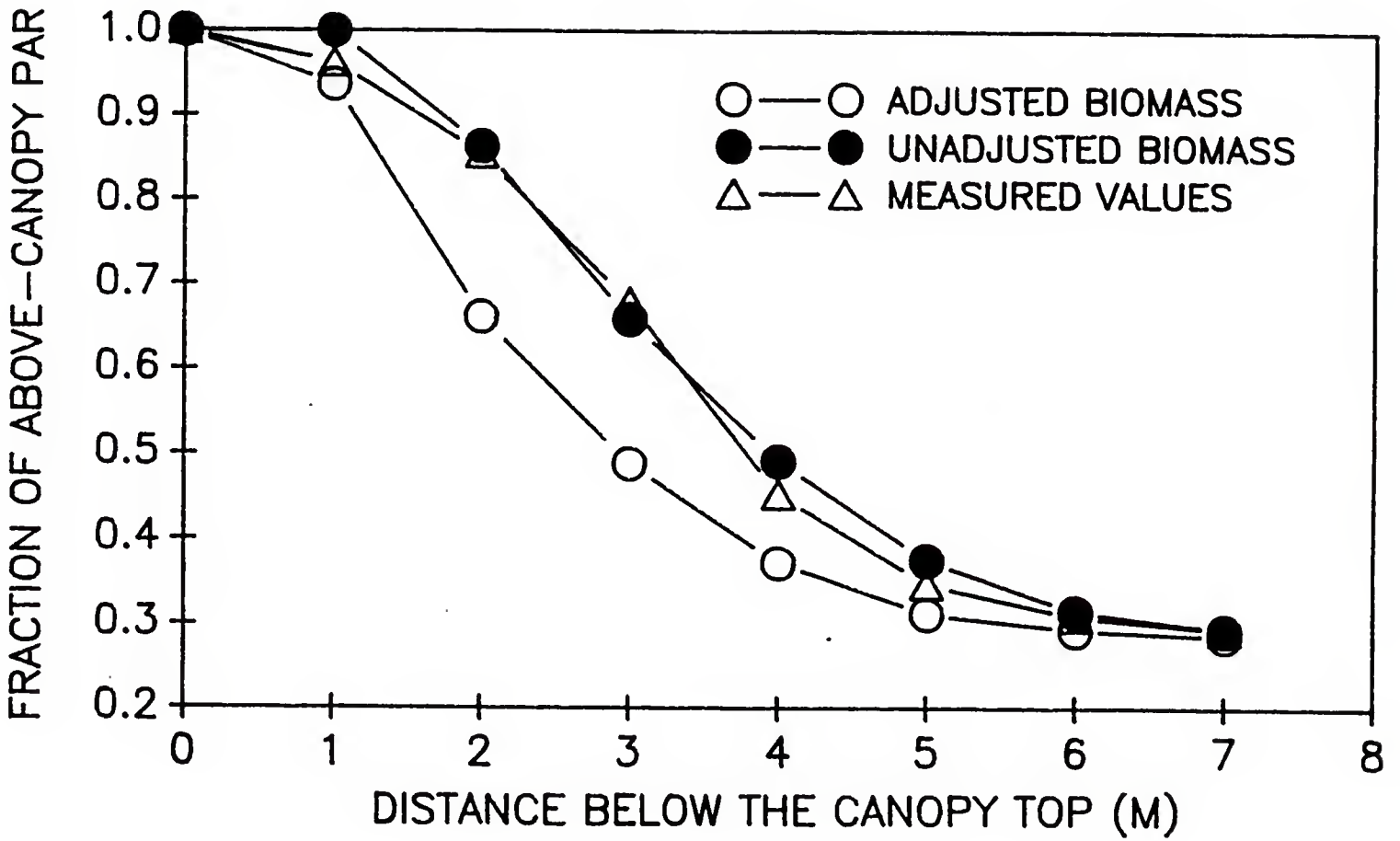

Figure 31: Stratified Beer's law simulations of light penetration through the canopy and average measured values.

EVALUATING THE MODEL

Cluster level modeling, up to this point, was parameterized by shifting the light penetration per cluster until a fit was obtained for the winter of 1987. Relative shifts in biomass over time were then used to generate light penetration patterns. When the light penetration per cluster was fit to the biomass estimates for the winter of 1987, the model runs predicted approximately 10% greater light penetration than was measured. There are several factors that can account for this discrepancy.

The model was sensitive to spacing of the trees. Small shifts in the spacing between trees, from 2.88 m to 3.2 m, caused a 10% (mean = 9.94%) shift in simulated light penetration (Figure 32). This change in spacing caused a 24% shift in leaf area, and a simple Beer's law model, matched to the initial light penetration, suggested that the shift should be about 7.6%. The difference between these model results lay in the conic form of the trees.

The difference in projected crown area due to the change in spacing, when viewed from 90 degrees AOE, was approximately 14%. Because the trees themselves had not

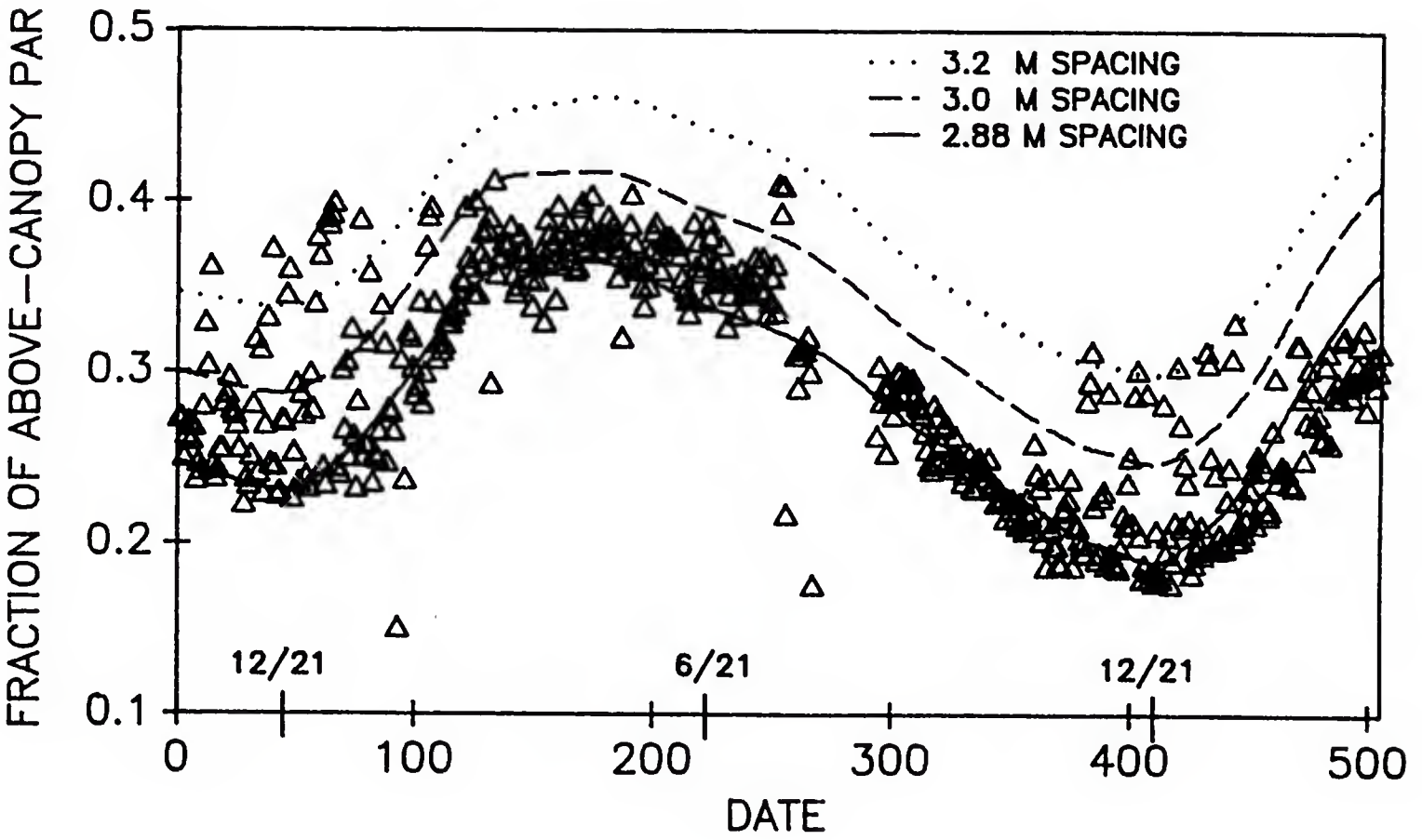

Figure 32: Simulation of the effect of tree spacing on light penetration through the canopy.

changed, the light penetration for vertical light also changed by 14%. In the simulation, none of the light emanated from 90 degrees AOE, and the relative increase in between-tree gaps was therefore less. The change in simulated light penetration through the canopy therefore lay between the estimate derived for a homogeneous stand and the value based on the penetration of vertical light.

The simulated crowns were probably narrower than the trees on which they were based. In the model, foliage was placed randomly within the foliated region of the branches. In fact, most of the foliage is clustered toward the end of the branch with an obligate cluster at the branch tip. The consequence of this difference is that, for any spacing, the gaps between the modeled trees were larger than the gaps through which the measured light was penetrating. This will bias the model toward higher light penetration for all foliage densities.

Shade caused by woody biomass is also not modeled. The branches and stem-wood did, however, shade the sensor arrays. If it is assumed that the trees are 17-m cones with basal diameters of 20 cm, each tree will project a maximum of 1.7 m of shadow. (Collection surface is normal to the light source.) If a tree density of 1120 stems per hectare is assumed, then the maximum stem area index (SAI) is

$$SAI_m = (1.7 * 1120)/1E4 = 0.19 \quad (25)$$

where

SAI_m = the maximum potential SAI.

Actual SAI will lie between this figure and the basal area index (BAI) which in this case would be = 3.5 E-3. The approximate calculation of SAI for a given sun angle is

$$SAI = SAI_m * \cos(AOE) \quad (26)$$

Complete branch data were not collected, but there are good estimates of the size and number of the primary branches (the longest continuous limbs). If the primary limbs project as triangles and are normal to the light, then the projection area will be

$$A_p = 1/2\ D_b\ L_b \quad (27)$$

where

A_p = the projected area,

D_b = the diameter at the base, and

L_b = length of the branch.

Based on this calculation, the estimated maximum total primary branch projected area is, for the 1987-1988

destructive samples, 0.51 m^2 per tree. Projecting this figure to a stand branch area index (BrAI), and assuming 1120 trees per hectare, BrAI is

$$BrAI = (0.51 * 1120)/1E4 = 0.058 \quad (28)$$

If the BrAI for the secondary limbs is assumed to be equal to the BrAI for the primary limbs, the total BrAI for the stand is approximately 0.12. Here again, the assumption of normalcy to the light is unrealistic. If it is assumed that total woody area index is half the potential maximum (0.15), then the total reduction in light anticipated from inclusion of branch area is

$$I_{bb} = I_{bf} \exp(-0.15) \quad (29)$$

where

I_{bb} = the light at the base of the branches, and

I_{bf} = the light at the base of the foliage.

If I_{bf} is 0.4, the inclusion of branch area will reduce this figure to 0.34. Inclusion of these estimates for woody biomass brings the simulated light penetration to within 4% of the curve that was manually fit to the light penetration data (Figure 19) at the base of the canopy and 2% at the forest floor (Figure 33).

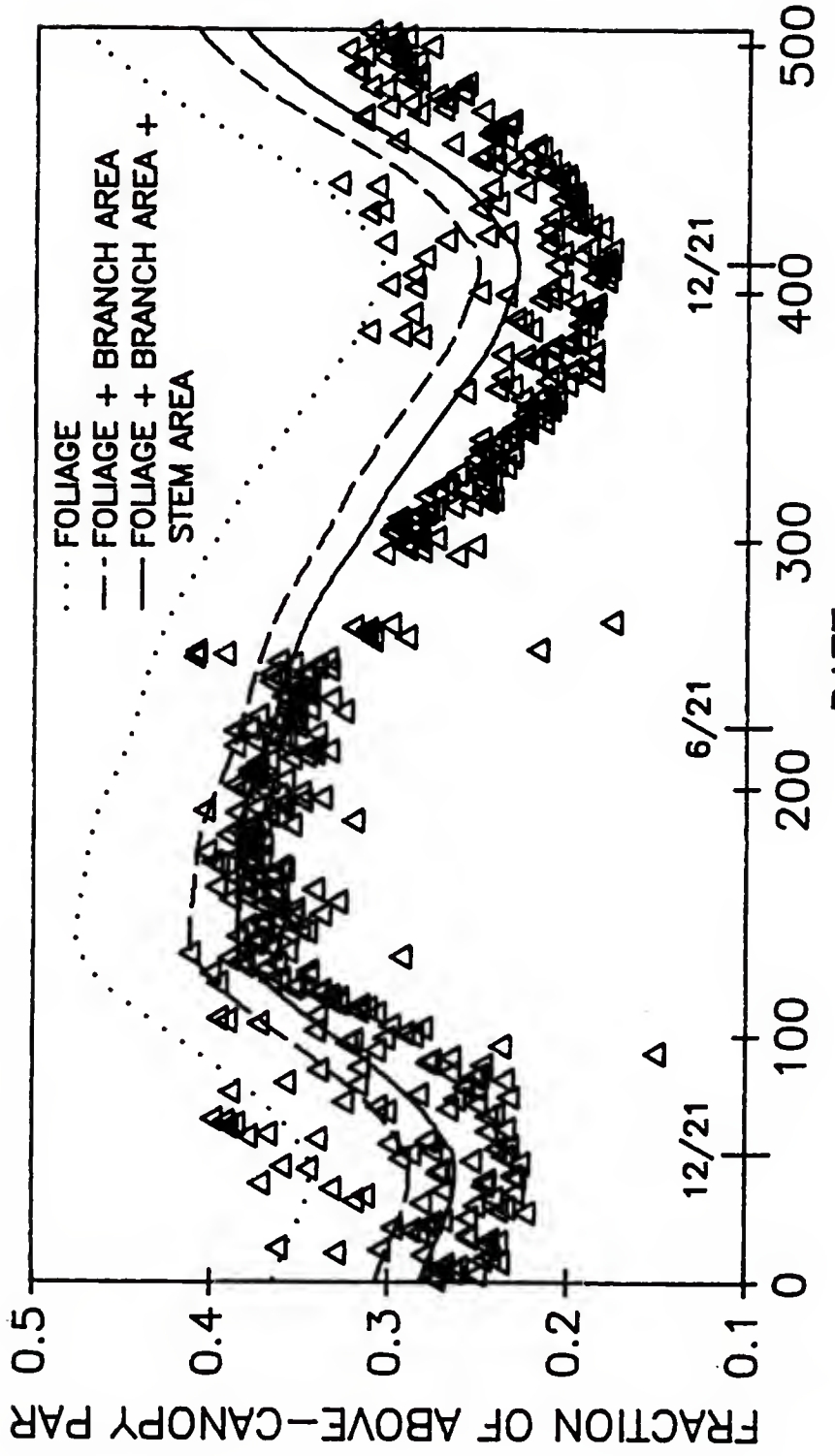

Figure 33: Simulation of the effects of woody biomass on below-canopy light penetration.

CONCLUSIONS

Discussion of Light Penetration through Slash Pine

Conifer canopies are composed of a hierarchy of discrete objects from the crown down to the needle. The higher levels of the hierarchy always have the potential to dominate light penetration patterns through the canopy. The extent of domination depends on the area density of the objects and their opacity. In this stand, structure at the crown and shoot levels does affect light penetration, though the effects are not as pronounced as theory suggests they might be.

The first hypothesis, that light penetration through the canopy will differ from a homogeneous canopy, is not falsified. The below-canopy light penetration pattern fits the cluster level model better than it fits a simple Beer's law model. The differences are, however, only seen for high AOE. The second hypothesis, that changes in between-tree spacing should have a stronger influence on light penetration than is anticipated from a similar shift in biomass in a homogeneous model, is, at least in the model,

also not falsified, though the effect is not large. The divergence of the cluster level model from a Beer's law model due to a shift in spacing from 2.88 m to 3.2 m is only 3%. The third hypothesis, that azimuth should be an important factor in light penetration, is not supported, at least for below-canopy light penetration. Light penetration does shift with azimuth, but, except for the unlikely occurrence of low sun-angle light passing directly down a row, the effects are unimportant. This may not be the case for all other stands, but will probably be valid for most closed canopies.

The ability of the cluster level model to simulate both below and within canopy light penetration demonstrates that expansion from clusters to canopies is possible, and that shifts in light penetration due to changes in biomass can be modeled effectively based on cluster level data. The cluster level model appears to be numerically stable. The similarity of its output to a homogeneous model based on cluster parameters suggests that the potential errors due to the assumption of uniform cluster properties and raster effects due to discrete representation have a negligible effect on estimations of light penetration. Output from the cluster level model, at least in a closed stand, simplifies to a single equation very nicely. The model can therefore be used to explore the spatial components of light

penetration through a particular stand, and, after the system is understood, the model output can be collapsed for modeling extended time series.

The approach to modeling diffuse light was very successful for simulating seasonal and daily patterns of variation in below-canopy light. Modeling diffuse light as light emanating from a fixed mid-sky location proved to be extremely effective. Coupled with a simple rule for partitioning direct and diffuse light, 85% of the variation in observed below-canopy light was accounted for over a two month period in the winter of 1987. Because winter is the time of the year when the system is most sensitive to the partitioning between direct and diffuse light, a partitioning rule can probably be created that will allow daily below-canopy light penetration to be simulated accurately throughout the year. In general, the models appear to be very powerful. Using a single equation to control light penetration, a curve for estimated biomass flux over time, and simple rules for producing light patterns on cloudy days, a rather complex pattern of light penetration can be matched, not only in its general form, but in its patterns of variation as well.

Both the model based on cluster level data and the simple Beer's law model fit the below-canopy light penetration patterns very well when changes in biomass were

included in the calculation. The validity of the technique used to estimate shifts in foliar biomass over time is supported by both models. This suggests that seasonal fluctuation in biomass can be estimated by inverting these models. The difficulty in matching light penetration to specific biomass estimates demonstrates that these models cannot be used to estimate leaf area directly, but can be used very effectively to track relative shifts in leaf area over time.

Overall, the results of cluster level modeling support the use of properly implemented stratified Beer's law models for closed-canopy stands of conifers. The model results do, however, raise the question of the biological relevance of planar light penetration modeling. If there is a great deal of self shading in the upper canopy at high sun angles, then estimates of photosynthesis or transpiration that are light-driven will be incorrect. Some self shade is inevitable in a species such as slash pine in which so much of the total foliage is concentrated in the narrow upper crown. Obtaining data to clarify this issue is extremely difficult. Cluster level modeling can provide estimates, and more sophisticated rules for cluster placement will improve these estimates, but defining the real operational environment of the foliage is still a distant goal.

The crowns of slash pine are thin and clumpy with well defined conic forms. Light penetration through slash pine canopies is therefore more likely to be influenced by canopy structure than the canopies of many other tree species. The fact that the effects of structure on below-canopy light penetration are very muted suggests that in most evenly spaced canopies, and particularly those at higher latitudes, canopy structure will be not be an important factor in determining average below-canopy light intensity. Canopy structure will, however, always have an important influence on light penetration through the canopy. Self shade in the dense upper canopy may be much greater for dense conic forms such as are characteristic of the true firs (Abies spp.).

The sensitivity of the model to differences in spacing suggests that treatments that modify spacing will have a greater impact on light penetration to the forest floor than may be anticipated based on the biomass removal. The cluster level model provides a useful tool to explore the relationships between specific spacing patterns and light penetration. This ability may prove useful for designing forest practices in which the manipulation of the understory vegetation is important: agro-forestry, controlling brush competition, and shelter-wood prescriptions.

Future Research Needs

To generalize the results derived here it will be necessary to determine whether the patterns of diffuse lighting are transferable to different plant architectures and time periods. This study showed that, for periods in which the incoming solar energy was similar, above-canopy PAR could be used to estimate the partitioning of diffuse and direct light. To be used throughout a growing season, the ratio of above canopy PAR to the theoretical maximum PAR (derived from the solar constant) will need to be used instead. The efficacy of this ratio for predicting light partitioning will need to be determined. Similarly, the ability to condense three-dimensional output to a single equation will probably not be possible for systems in which azimuth is an important parameter for determining light penetration patterns. For these systems, light penetration patterns generated by the three dimensional model may need to be used to simulate each time step.

LITERATURE CITED

Campbell, G.S., 1977. An introduction to environmental biophysics. Springer-Verlag, New York

Campbell, G.S.; Norman, J.M. 1989. The description and measurement of plant canopy structure. In: Plant canopies: their growth, form and function. Russell, G.; Marshall, B; Jarvis, P.G. Editors. Cambridge University Press, Cambridge. 1-20.

Carter, G.A.; Smith, W.K. 1985. Influence of shoot structure on light interception and photosynthesis in conifers. Plant Physiology, 79: 1038-1043.

Charles-Edwards, D.A.; Thorpe, M.R., 1976. Interception of diffuse and direct-beam radiation by a hedgerow apple orchard. Annals of Botany, 40: 603-613.

Cohen, S.; Fuchs, M. 1987. The distribution of leaf area, radiation, photosynthesis and transpiration in a shamouti orange hedgerow orchard. Part I. Leaf area and radiation. Agricultural and Forest Meteorology. 40: 123-144.

Gates, D. M. 1980. Biophysical ecology. Springer Verlag, New York.

Gholz, H.L.; Fisher, R.F. 1982. Organic matter production and distribution in slash pine (Pinus elliottii) plantations. Ecology, 63:1827-1839.

Gholz, H.L.; Vogel, S.A.; Cropper, W.P.; McKelvey, K.; Ewel, K.C; Teskey, R.O.; Curran, P.J. 1990. Dynamics of canopy structure and light interception in Pinus elliottii stands of north Florida. Ecology, in press.

Grace, J. C. 1988. Effect of tree arrangement on intercepted radiant energy and photosynthesis. In: Forest growth and modelling prediction, Volume I. Ek, A.R; Shifley, S.R.; Burk, T.E. Editors. USDA, Forest Service, General Technical Report NC-120. 285-292.

Grant, R.H., 1985. The influence of the sky radiance distribution on the flux density in the shadow of a tree crown. Agricultural and Forest Meteorology, 35: 59-70.

Lang, A.R.G.; Yuequin, X. 1986. Estimation of leaf area index from transmission of direct sunlight in discontinuous canopies. Agricultural and Forest Meteorology, 37: 229-243.

Mann, J.E.; Curry, G,L; DeMichele, D.W.; Baker, D.N. 1980. Light penetration in a row crop with random plant spacing. Agronomy Journal, 72: 131-142.

Mann, J.E.; Curry, G.L.; Sharpe,P.J.H. 1979. Light interception by isolated plants. Agricultural Meteorology, 20: 205-214.

McKelvey, K.S. 1988. A geometric model of sunlight penetration for slash pine in northern Florida. In: Forest growth and modelling prediction, Volume I. Ek, A.R; Shifley, S.R.; Burk, T.E. Editors. USDA Forest Service, General Technical Report NC-120, 323-330.

Norman, J.M.; Welles, J.M. 1983. Radiative transfer in an array of canopies. Agronomy Journal, 75: 481-488.

Oker-Blom, P. 1983. Effect of grouping of foliage on the within-stand and within-crown light regime: comparisons of random and grouping canopy models. Agricultural Meteorology, 28: 143-155

Oker-Blom, P. 1986. Photosynthetic radiation regime and canopy structure in modeled forest stands. Acta Forestalia Fennica, Volume 197.

Pierce, L.L.; Running, S.W. 1988. Rapid estimation of coniferous forest leaf area index using a portable integrating radiometer. Ecology, 69: 1762-1767.

Rook, D.A.; Grace, J.C.; Beets, P.N.; Whitehead, D.; Santantonio, D.; Madgewick, H.A.I. 1985. Forest canopy design: Biological models and management implications. In: Attributes of trees as crop plants. M.G.R. Cannel and J.E. Jackson editors, 507-524.

Spitters, C.J.T.; Toussaint, H.A.J.M; Goudriaan, J. 1986. Separating the diffuse and direct component of global radiation and its implications for modeling canopy

photosynthesis. Part I. Components of incoming radiation. Agricultural and Forest Meteorology, 38: 217-229.

Walraven, R. 1978. Calculating the position of the sun. Solar Energy, 20:393-397.

Whitfield, D.M., 1986. A simple model of light penetration into row crops. Agricultural and Forest Meteorology, 36: 297-315

APPENDIX

COMPUTER CODE FOR DISCRETE-SPACE MODELING OF LIGHT PENETRATION

```
{Included here is all of code required to rotate the targets
and a simple calling program.  The program is built in
modular form according to the rules in Turbo Pascal V. 5.0
(Turbo Pascal is a trademark of Borland International).
Several other modules are mentioned but not listed. They are
all standard libraries in Turbo Pascal 5.0 with the
exception of Math, which contains power functions and
transcendentals.  Math is of no scientific interest and is
not included here.  The modules are listed in order of their
dependencies: Global is first because all of the others
require it, and the calling program is last.}

                    (*****UNIT GLOBAL*****)

{$N+,E+,V-,S-}

{Global contains all of the global variables, types and
constants. It also includes two procedures of general
utility}

UNIT Global;

INTERFACE

USES Crt,Dos;

{Global types and constants}

CONST
    Max_Pts       = 10000;
    Select_Pts    = 5000;
    Max_Targets   = 3000;
    Max_Data_Pts  = 3;
    Observations  = 1;
    NoCursor      = $2000;

{these used in graphics; gd = graphics driver and is set to
detect mode}
```

```
{gm = graphmode}

    Gd              : Integer = 0;
    Gm              : Integer = 0;
    Scale_By        : Single = 1;

TYPE
    String80         = String[80];
    CharSet          = Set OF Char;
    TargetDim        = (MinX,MaxX,MinY,MaxY,MinZ,MaxZ);
    IntCoordinates   = Array[1..4]              OF Integer;
    IntCoordArr      = Array[0..Select_Pts]     OF
IntCoordinates;
    IntTargetArr     = Array[0..Max_Targets]    OF
IntCoordinates;
    IntArrType       = Array[0..Max_Pts]        OF Integer;
    ArrType          = Array[0..Max_Pts]        OF Single;
    IntSelectType    = Array[0..Select_Pts]     OF Integer;
    SelectType       = Array[0..Select_Pts]     OF Single;
    Matrix3d         = Array[1..4,1..4]         OF Single;
    TargetMaxMin     = Array[TargetDim]         OF Single;
    ShadeArrType     = Array[1..Observations]   OF Single;
    ClusterData      = RECORD
                             ShadeArr      : ShadeArrType;
                             NotFirstTime  : Boolean;
                       END;
    ClusterArr        = Array[1..Max_Targets] OF ClusterData;
    IntTargetGpharr   = Array[0..Max_Targets] OF Word;
    Targetarr         = Array[0..Max_Targets] OF Single;

    GraphCoordArr     = Array[1..Max_Targets,1..2] OF Single;
    IntGraphCoordArr  = Array[1..Max_Targets,1..2] OF Word;
    DataArray         = Array[0..Max_Data_Pts] of single;

    ArrPtr           = ^ArrType;
    IntArrPtr        = ^IntArrType;
    SelectPtr        = ^SelectType;
    IntSelectPtr     = ^IntSelectType;
    IntTargetPtr     = ^IntTargetGphArr;
    TargetPtr        = ^TargetArr;
    DataPtr          = ^DataArray;


{Global static variables}
VAR

{these are primarily for LIBRARY}

   ClusterInfo     : ClusterArr;
```

```
{TargetCenter... are produced in Menu and used generally}

   TargetCenterX,
   TargetCenterY,
   TargetCenterZ  : Single;

{these represent the basic data input}

BaseX,BaseZ,
Az,AOE,ACIntensity              : DataPtr;
Row,Col,TargetNo,TotalTrees,
TargetRow,TargetCol,TotalObs,
Day,Month                       : Integer;
TotalClusters                   : Word;
RowDim,ColDim,Xdim,Zdim,
TargetPosX,TargetPosZ,LoAOE,
HiAOE,ConstAz,ConstAOE,
LoAz,HiAz,MinAOE,Latitude       : Single;

{Global heap ptrs}

ExitVect                        : Pointer;
Xd,Yd,Zd,LightPen               : ArrPtr;
IdVect                          : IntArrPtr;
TempX,TempY,TempZ               : SelectPtr;
TempId                          : IntSelectPtr;
TempX2,TempY2,TempZ2            : SelectPtr;
TempId2                         : IntSelectPtr;
TX,TY,TZ,Sum                    : Targetptr;
Id                              : Inttargetptr;
GraphHeapPoint                  : Pointer;
HeapPtr1,HeapPtr2               : Pointer;

{Global Procedures and Functions}

PROCEDURE FatalErrors(ErrSt : string80);

IMPLEMENTATION

{setcursor uses a DOS interrupt to set the cursor either on
 or off}

PROCEDURE FatalErrors(ErrSt : string80);
          BEGIN
               gotoxy(10,12);
               Write(ErrSt);
               Delay(2000);
               Halt;
END;
BEGIN
```

```
END.

                    (*****UNIT ARATIO*****)

{ARatio determines whether a particular group of points
should be clipped from the shade search. The algorithm will
work without it but will be slower.}

UNIT ARatio;

INTERFACE

USES Math,Global;

PROCEDURE GetMaxVal;
FUNCTION AOESearchSw(Azimuth,AngOfElevation : Single):
         Boolean;

IMPLEMENTATION

CONST
     TreeRad = 1;
     PiDiv2  = 1.5709763;
     Percent = 0.35;
VAR
   MaxDimX,MaxDimZ,MinDimX,
   MinDimZ,TargetDimX,TargetDimZ : Single;

{FindMaxDim searches the base arrays and finds the edges and
 the targets. These positions are scaled by an arbitrary
amount representing the approximate radius of the trees}

PROCEDURE FindMaxDim(X,Z                    : DataPtr;
                     TotalArrSz,
                     TargetArrPos           : Byte;
                     TreeRad                : Single;
                 VAR MaxDimX,MaxDimZ,
                     TargetDimX,TargetDimZ,
                     MinDimX,MinDimZ        : Single);
          VAR
             Ct          : Byte;
          BEGIN
               MaxDimX := -9E9;
               MaxDimZ := -9E9;
               MinDimZ :=  9E9;
               MinDimX :=  9E9;
               FOR Ct := 1 TO TotalArrSz DO BEGIN
                   IF X^[Ct] > MaxDimX THEN
                      MaxDimX := X^[Ct];
                   IF X^[Ct] < MinDimX THEN
```

```
                    MinDimX := X^[Ct];
                IF Z^[Ct] > MaxDimZ THEN
                    MaxDimZ := Z^[Ct];
                IF Z^[Ct] < MinDimZ THEN
                    MinDimZ := Z^[Ct];
            END;
            TargetDimX := X^[TargetArrPos];
            TargetDimZ := Z^[TargetArrPos];
            MaxDimX := MaxDimX + TreeRad;
            MaxDimZ := MaxDimZ + TreeRad;
            MinDimZ := MinDimZ - TreeRad;
            MinDimX := MinDimX - TreeRad;
END;

{FindMaxPos finds the distance across the system by assuming
 a rectangular array of trees.  Distance computed is the
hypotenuse}

FUNCTION FindMaxPos(MaxX,MaxZ,TargetX,TargetZ,Azimuth :
          Single) : Single;
      VAR
         CritAng,MaxDist,LenX,LenZ : Single;
      BEGIN
           LenX := MaxX - TargetX;
           LenZ := MaxZ - TargetZ;
           CritAng := ArcTan(LenZ/LenX);
           Azimuth := Azimuth * DegToRad;

{now that we have defined the corner (maxX,maxZ), if the
angle is > than the angle defined by that corner and the
target tree, we use the complement to find the hypotenuse}

           IF Azimuth > CritAng THEN BEGIN
              Azimuth := PiDiv2 - Azimuth;   {complement}
              FindMaxPos := LenZ/Cos(Azimuth);
           END
           ELSE
               FindMaxPos := LenX/Cos(Azimuth);
END;

{Searching for the Az vector is always considered to be
 appropriate because of the generally square nature of most
 point sets. This means that a large number of points will
be removed in an azimuth search.  The efficiency of an AOE
search depends on the angle and on the path through the
system that a vector with that angle must pass. This
algorithm makes the decision to search or not based on the
 asumption of a homogenous point set. 30% is the cutoff
because the search is cheaper numerically than the
subsequent rotation and shade search.}
```

```
FUNCTION AOESw(Len,TotalHt,AOE : Single): Boolean;
         VAR
            TanAOE,TotalArea,
            TriangleHt,TriangleLen,
            TriangleArea,Additionalarea      : Single;
         BEGIN
              IF Len < 1.0 THEN BEGIN      {not worth doing}
                 AOESw := FALSE;
                 Exit;
              END;
              TotalArea := Len * TotalHt;
              TanAOE := Tan(DegToRad * AOE);{a holder}
              TriangleHt := TanAOE * Len;
              IF TriangleHt > TotalHt THEN BEGIN {triangular
                                                  area and}
                 TriangleLen := TotalHt/TanAOE;  {square
                                                 area
                                                  removed}
                 AdditionalArea := (Len - TriangleLen) *
                                   TotalHt;
                 TriangleArea := TriangleLen * TotalHt *
                                 0.5;

       {Search switch is thrown by an area removal of
 greater than 30%}

                 AOESw := ((TriangleArea +
                          AdditionalArea)/TotalArea >
                          Percent);
              END
              ELSE BEGIN
                   AOESw := ((TriangleHt * Len *
                0.5)/TotalArea > Percent);
              END;
END;
PROCEDURE GetMaxVal;
         BEGIN

FindMaxDim(BaseX,BaseZ,TotalTrees,TargetNo,TreeRad,
           MaxDimX,MaxDimZ,TargetDimX,TargetDimZ,
           MinDimX,MinDimZ);
END;

FUNCTION AOESearchSw(Azimuth,AngOfElevation : Single):
   Boolean;
         BEGIN
              IF Azimuth  < 0 THEN
           AOESearchSw:=AOESw(FindMaxPos(MaxDimX,
                              TargetDimZ - MinDimZ,
```

```
                                0,0,Abs(Azimuth)),
                                Depth,AngOfElevation)
                ELSE
                 AOESearchSw := AOESw(FindMaxPos(
                                      MaxDimX,
                                      MaxDimZ,
                                      TargetDimX,
                                      TargetDimZ,
                                      Azimuth),
                                      Depth,
                                      AngOfElevation);
END;

BEGIN
END.
```

```
                    (*****UNIT LIBRARY*****)

{Unit library contains the rotation and shade search
algorithms}

UNIT Library;

INTERFACE

USES Crt,Math,Global,ARatio;

{Global Procedures}

PROCEDURE FindMaxMin(X,Y,Z          : ArrPtr;
                     Targets        : Word;
                 VAR DefineTarget   : TargetMaxMin);
PROCEDURE GetShadePts(X,Y,Z              :ArrPtr;
                      Id                 : IntArrPtr;
                      TotalPts,Targets   : Word;
                      AOE,Azimuth        : Single;
                      CenterX,CenterY,
                      CenterZ            : Single;
                      DefineTarget       : TargetMaxMin;
                      ObsNo              : Byte;
                      SelfShade,NewSim   : Boolean);


IMPLEMENTATION
CONST
     Identity3d : Matrix3d =
((1,0,0,0),(0,1,0,0),(0,0,1,0),(0,0,0,1));
VAR
   M11,M21,M31,M41,M12,
```

```
   M22,M32,M42,M13,M23,M33,M43 : Single;

{WordMove uses MOVSW rather than MOVSB and achieves a 37%
speedup - NOTE-8086 assembly language}

procedure WordMove(var Source,Dest;Number : word);
begin
     inline($1E                    {push ds}
           /$C5/$B6/SOURCE
           /$C4/$BE/DEST
           /$8B/$8E/NUMBER
           /$D1/$E9                {shr cx,1}
           /$FC
           /$F3/$A5                {rep stosw}
           /$1F);                  {pop ds}
END;

{These procedures determine the number of points that are
potential shading objects for the target. FindMaxMin creates
a "cube" around the target that defines its boundaries.
ClipByAz creates lines parallel to the solar azimuth
that bound the target in the Z dimension.  The reduced set
of points is then Clipped in the Y (vertical) dimension by
creating lines parallel with the solar AOE.  The remaining
points are then transferred into the static arrays for
rotation and shade search.  Since the targets are always in
the selected group, they are transferred in the MoveTargets
procedure and their positions are not calculated in ClipByAz
orClipBy AOE}

PROCEDURE FindMaxMin(X,Y,Z          : ArrPtr;
                     Targets        : Word;
                 VAR DefineTarget : TargetMaxMin);
          VAR
               Ct : Word;
          BEGIN
               DefineTarget[MinX] := 9E9;
               DefineTarget[MinY] := 9E9;
               DefineTarget[MinZ] := 9E9;
               DefineTarget[MaxX] := -9E9;
               DefineTarget[MaxY] := -9E9;
               DefineTarget[MaxZ] := -9E9;
               FOR Ct := 1 TO Targets DO BEGIN
                   IF X^[Ct] < DefineTarget[MinX] THEN
                      DefineTarget[MinX] := X^[Ct];
                   IF X^[Ct] > DefineTarget[MaxX] THEN
                      DefineTarget[MaxX] := X^[Ct];
                   IF Y^[Ct] < DefineTarget[MinY] THEN
                      DefineTarget[MinY] := Y^[Ct];
                   IF Y^[Ct] > DefineTarget[MaxY] THEN
```

```
                    DefineTarget[MaxY] := Y^[Ct];
                IF Z^[Ct] < DefineTarget[MinZ] THEN
                    DefineTarget[MinZ] := Z^[Ct];
                IF Z^[Ct] > DefineTarget[MaxZ] THEN
                    DefineTarget[MaxZ] := Z^[Ct];
            END;
END;
PROCEDURE ClipByAz(Azimuth        : Single;
                   BeginPt,
                   PointNo        : Integer;
                   X,Y,Z          : ArrPtr;
                   Id             : IntArrPtr;
                   TempX,
                   TempY,TempZ    : SelectPtr;
                   TempId         : IntSelectPtr;
                   DefineTarget   : TargetMaxMin;
                VAR PotShade      : Integer);
        VAR
           Zslope,ZHi,ZLo,ZintHi,ZintLo,Diff : Extended;
           AddForSafety                      : Extended;
           Ct1                               : Integer;
        BEGIN
            Zslope := Tan(DegToRad * Azimuth);
            AddForSafety := 1.0/Cos(Azimuth * DegToRad);

            {Addition of a safety buffer by shifting the
             intercept}
            {far enough to move the lines +- 1.0 unit in
             the direction}
            {normal to the slope}

            ZintLo := (DefineTarget[MinZ] - Zslope *
                       DefineTarget[MaxX]) - AddForSafety;
            ZintHi := (DefineTarget[MaxZ] - Zslope *
                       DefineTarget[MinX]) + AddForSafety;
            Diff := ZintHi - ZintLo;
            PotShade := BeginPt;
            FOR Ct1 := Succ(BeginPt) TO PointNo DO BEGIN

                {start after target pts.}
                {define a band from the lowest to
          highest, inclusive}

                ZHi := X^[Ct1] * Zslope + ZintHi;
                IF (Z^[Ct1] <= ZHi) AND (Z^[Ct1] >=(ZHi -
                   Diff)) THEN
                   BEGIN
                   Inc(PotShade);
                   IF PotShade = Select_Pts THEN
                      FatalErrors('Too many shade points,
```

```
                                        Program halted');
                    TempX^[PotShade]  := X^[Ctl];
                    TempY^[PotShade]  := Y^[Ctl];
                    TempZ^[PotShade]  := Z^[Ctl];
                    TempId^[PotShade] := Id^[Ctl];
                END;
            END;
END;
PROCEDURE ClipByAOE(AOE             : Single;
                    BeginPt,
                    PointNo         : Integer;
                    X,Y,Z           : SelectPtr;
                    Id              : IntSelectPtr;
                    TX,TY,TZ        : SelectPtr;
                    TId             : IntSelectPtr;
                    DefineTarget    : TargetMaxMin;
             VAR PotShade           : Integer);
        VAR
           Yslope,YLo,YintLo,Diff                : Extended;
           AddForSafety                          : Extended;
           Ctl                                   : Integer;
        BEGIN
            Yslope := Tan(DegToRad * AOE);
            AddForSafety := 1.0/Cos(AOE * DegToRad);

            {Addition of a safety buffer by shifting the
             intercept}
            {far enough to move the line +- 1.0 unit in
            the direction}
            {normal to the slope}

            YintLo := (DefineTarget[MinY] - Yslope *
                       DefineTarget[MaxX]) - AddForSafety;
            PotShade := BeginPt;
            FOR Ctl := Succ(BeginPt) TO PointNo DO BEGIN

                {start after target pts.}
                {define a band from the lowest to
          highest, inclusive}

                YLo := X^[Ctl] * Yslope + YintLo;
                IF Y^[Ctl] >= YLo THEN BEGIN
                   Inc(PotShade);
                   TX^[PotShade] := X^[Ctl];
                   TY^[PotShade] := Y^[Ctl];
                   TZ^[PotShade] := Z^[Ctl];
                   TId^[PotShade] := Id^[Ctl];
                END;
            END;
END;
```

```
PROCEDURE MoveTargets(Targets  : Word; X,Y,Z : ArrPtr;
                      Id       : IntArrPtr;
                      Tx,Ty,Tz : SelectPtr;
                      Id2      : IntSelectPtr);
          VAR
             ByteNo : Word;
       {here we move the target points into the temporary and
        final arrays}
          BEGIN
               Inc(Targets);
               ByteNo := Targets * SizeOf(Single);
               {single = 6 bytes if 80x87 absent}
                WordMove(X^,TX^,ByteNo);
                WordMove(Y^,TY^,ByteNo);
                WordMove(Z^,TZ^,ByteNo);
                WordMove(Id^,Id2^,Targets * 2);
END;


{These Procedures perform the various sorting tasks.}

PROCEDURE SelectIntSort(n: Integer;VAR Arr : IntCoordArr);
LABEL JumpToQsort,JumpToSortStack,Fin;   {Y sort chosen}
CONST
   SelectInsertSort = 7;
   SortStackTop     = 20;
VAR
   Top,Bottom,StackMarker,
   Ct1,Ct2,RandomStart          : Integer;
   First,Second,Third,Fourth    : Integer;
   SortStackArr                 : Array[1..SortStackTop] OF
Integer;
BEGIN
 IF n > 1 THEN BEGIN
   StackMarker := 0;
   Bottom      := 1;
   Top         := n;
   WHILE TRUE DO BEGIN
      IF ((Top-Bottom) < SelectInsertSort) THEN BEGIN
         FOR Ct1 := Succ(Bottom) TO Top DO BEGIN
            First  := Arr[Ct1,1];
            Second := Arr[Ct1,2];
            Third  := Arr[Ct1,3];
            Fourth := Arr[Ct1,4];
            Ct2    := Pred(Ct1);
            WHILE (Arr[Ct2,2] > Second) AND (Ct2 > 0) DO
BEGIN
               Arr[Succ(Ct2)] := Arr[Ct2];
               Dec(Ct2);
            END;
```

```
        Inc(Ct2);
        IF Ct2 <> Ct1 THEN BEGIN
            Arr[Ct2,1] := First;
            Arr[Ct2,2] := Second;
            Arr[Ct2,3] := Third;
            Arr[Ct2,4] := Fourth;
         END;
    END;
    IF (StackMarker = 0) THEN GOTO Fin;
    Top := SortStackArr[StackMarker];
    Bottom := SortStackArr[Pred(StackMarker)];
    Dec(StackMarker,2);
 END ELSE BEGIN
    Ct2 := Bottom;
    Ct1 := Top;
    RandomStart := RANDOM(Top - Bottom) + Bottom;
    First  := Arr[RandomStart,1];
    Second := Arr[RandomStart,2];
    Third  := Arr[RandomStart,3];
    Fourth := Arr[RandomStart,4];
    Arr[RandomStart] := Arr[Bottom];
JumpToQsort:
    WHILE (Second < Arr[Ct1,2]) AND (Ct1 > 0) DO
          Dec(Ct1);
    IF (Ct1 <= Ct2) THEN BEGIN
       Arr[Ct2,1] := First;
       Arr[Ct2,2] := Second;
       Arr[Ct2,3] := Third;
       Arr[Ct2,4] := Fourth;
       GOTO JumpToSortStack
    END;
    Arr[Ct2] := Arr[Ct1];
    Inc(Ct2);
 WHILE (Second > Arr[Ct2,2]) AND (Ct2 <= n) DO
       Inc(Ct2);
    IF (Ct1 <= Ct2) THEN BEGIN
       Arr[Ct1,1] := First;
       Arr[Ct1,2] := Second;
       Arr[Ct1,3] := Third;
       Arr[Ct1,4] := Fourth;
       Ct2 := Ct1;
       GOTO JumpToSortStack
    END;
    Arr[Ct1] := Arr[Ct2];
    Dec(Ct1);
    GOTO JumpToQsort;
JumpToSortStack:
    Inc(StackMarker,2);
    IF ((Top-Ct2) >= (Ct2-Bottom)) THEN BEGIN
       SortStackArr[StackMarker]       := Top;
```

```
            SortStackArr[Pred(StackMarker)] := Succ(Ct2);
            Top := Pred(Ct2)
         END ELSE BEGIN
            SortStackArr[StackMarker]       := Pred(Ct2);
            SortStackArr[Pred(StackMarker)] := Bottom;
            Bottom := Succ(Ct2)
         END
      END
   END;
Fin:
END;
END ;

PROCEDURE TargetIntSort(n: Integer;VAR Arr : IntTargetArr);
LABEL JumpToQsort,JumpToSortStack,Fin;                    {Y
sort chosen}
CONST
   SelectInsertSort = 7;
   SortStackTop     = 20;
VAR
   Top,Bottom,StackMarker,
   Ct1,Ct2,RandomStart          : Integer;
   First,Second,Third,Fourth    : Integer;
   SortStackArr                 : Array[1..SortStackTop] OF
Integer;
BEGIN
 IF n > 1 THEN BEGIN
   StackMarker := 0;
   Bottom      := 1;
   Top         := n;
   WHILE TRUE DO BEGIN
      IF ((Top-Bottom) < SelectInsertSort) THEN BEGIN
         FOR Ct1 := Succ(Bottom) TO Top DO BEGIN
            First  := Arr[Ct1,1];
            Second := Arr[Ct1,2];
            Third  := Arr[Ct1,3];
            Fourth := Arr[Ct1,4];
            Ct2    := Pred(Ct1);
            WHILE (Arr[Ct2,2] > Second) AND (Ct2 > 0) DO
BEGIN
               Arr[Succ(Ct2)] := Arr[Ct2];
               Dec(Ct2);
            END;
            Inc(Ct2);
            IF Ct2 <> Ct1 THEN BEGIN
                Arr[Ct2,1] := First;
                Arr[Ct2,2] := Second;
                Arr[Ct2,3] := Third;
                Arr[Ct2,4] := Fourth;
             END;
```

```
    END;
    IF (StackMarker = 0) THEN GOTO Fin;
    Top := SortStackArr[StackMarker];
    Bottom := SortStackArr[Pred(StackMarker)];
    Dec(StackMarker,2);
 END ELSE BEGIN
    Ct2 := Bottom;
    Ct1 := Top;
    RandomStart := RANDOM(Top - Bottom) + Bottom;
    First  := Arr[RandomStart,1];
    Second := Arr[RandomStart,2];
    Third  := Arr[RandomStart,3];
    Fourth := Arr[RandomStart,4];
    Arr[RandomStart] := Arr[Bottom];
JumpToQsort:
    WHILE (Second < Arr[Ct1,2]) AND (Ct1 > 0) DO
          Dec(Ct1);
    IF (Ct1 <= Ct2) THEN BEGIN
       Arr[Ct2,1] := First;
       Arr[Ct2,2] := Second;
       Arr[Ct2,3] := Third;
       Arr[Ct2,4] := Fourth;
       GOTO JumpToSortStack
    END;
    Arr[Ct2] := Arr[Ct1];
    Inc(Ct2);
 WHILE (Second > Arr[Ct2,2]) AND (Ct2 <= n) DO
       Inc(Ct2);
    IF (Ct1 <= Ct2) THEN BEGIN
       Arr[Ct1,1] := First;
       Arr[Ct1,2] := Second;
       Arr[Ct1,3] := Third;
       Arr[Ct1,4] := Fourth;
       Ct2 := Ct1;
       GOTO JumpToSortStack
    END;
    Arr[Ct1] := Arr[Ct2];
    Dec(Ct1);
    GOTO JumpToQsort;
JumpToSortStack:
    Inc(StackMarker,2);
    IF ((Top-Ct2) >= (Ct2-Bottom)) THEN BEGIN
       SortStackArr[StackMarker]       := Top;
       SortStackArr[Pred(StackMarker)] := Succ(Ct2);
       Top := Pred(Ct2)
    END ELSE BEGIN
       SortStackArr[StackMarker]       := Pred(Ct2);
       SortStackArr[Pred(StackMarker)] := Bottom;
       Bottom := Succ(Ct2)
    END
```

```
      END
   END;
Fin:
END;
END ;

{These procedures create the rotation matrix, rotate the
coordinates and search for shade.  MatrixMultiply multiplies
two 4x4 matrices.  Matrix VectorMultiply multiplies a
coordinate vector (X,Y,Z) with a 3x4 matrix
to transform it. ProduceRotationMatrix creates the rotation
matrix. All of the other procedures (CreateYRotationMatrix
etc.) perform the various steps necessary to create a matrix
that will transform a series of coordinates so that the
points defined by those coordinates rotate around an
arbitrary X,Y,Z location.  Details of transformation
matrices can be found in a variety of introductory Computer
Graphics texts}

PROCEDURE MatrixMultiply(VAR M1 : Matrix3d;VAR M2 :
Matrix3d);
          VAR
             Row,Column          : Integer;
             TempMat             : Matrix3d;
          BEGIN
            FOR Row := 1 TO 4 DO
                FOR Column := 1 TO 4 DO
                   TempMat[Row,Column] := M1[Row,1] *
                                          M2[1,Column] +
                                          M1[Row,2] *
                                          M2[2,Column] +
                                          M1[Row,3] *
                                          M2[3,Column] +
                                          M1[Row,4] *
                                     M2[4,Column];
                    M1 := TempMat;
END;
PROCEDURE VectorMatrixMultiply(Points : Word;X,Y,Z :
SelectPtr);
          VAR
             TempX,TempY     : Single;
             Ct              : Word;
          BEGIN
              FOR Ct := 1 TO Points DO BEGIN
                  TempX := X^[Ct] * M11 + Y^[Ct] * M21 +
                           Z^[Ct] * M31 + M41;
                  TempY := X^[Ct] * M12 + Y^[Ct] * M22 +
                           Z^[Ct] * M32 + M42;
                  Z^[Ct] := X^[Ct] * M13 + Y^[Ct] * M23 +
                            Z^[Ct] * M33 + M43;
```

```
                X^[Ct] := TempX;
                Y^[Ct] := TempY;
            END;
END;
PROCEDURE CreateTranslation(X,Y,Z : Single; VAR T :
Matrix3d);
          BEGIN
               T := Identity3d;
               T[4,1] := X;
               T[4,2] := Y;
               T[4,3] := Z;
END;
PROCEDURE CreateYRotationMatrix(    X1,Y1,Z1,Angle
: Single;
                                    VAR TempMat
: Matrix3d);
          VAR
             TempAng,Sina,
             Cosa              : Extended;
             M                 : Matrix3d;


{start rotation procedure}

BEGIN

{translate}

    CreateTranslation(-X1,-Y1,-Z1,TempMat);

{do Y rotation}

    TempAng := -DegToRad * Angle;
    Sina    := Sin(TempAng);
    Cosa    := Cos(TempAng);
     M      := Identity3d;
     M[1,1] := Cosa;
     M[1,3] := -Sina;
     M[3,1] := Sina;
     M[3,3] := Cosa;
   MatrixMultiply(TempMat,M);

  {inverse translation}

     CreateTranslation(X1,Y1,Z1,M);
     MatrixMultiply(TempMat,M);
END;
PROCEDURE CreateZRotationMatrix(X1,Y1,Z1,Angle : Single;
                                VAR TempMat      : Matrix3d);
          VAR
```

```
            TempAng,Sina,
            Cosa              : Extended;
            M                 : Matrix3d;

{start rotation procedure}

BEGIN

{translate}

    CreateTranslation(-X1,-Y1,-Z1,TempMat);

{do Z rotation}

    TempAng := DegToRad * Angle;
    Sina    := Sin(TempAng);
    Cosa    := Cos(TempAng);
     M      := Identity3d;
     M[1,1] := Cosa;
     M[1,2] := Sina;
     M[2,1] := -Sina;
     M[2,2] := Cosa;
   MatrixMultiply(TempMat,M);

  {inverse translation}

     CreateTranslation(X1,Y1,Z1,M);
     MatrixMultiply(TempMat,M);
END;
PROCEDURE Rotate(CenterX,CenterY,CenterZ,Azimuth,AOE :
Single);
                  VAR
                     Trans1,Trans2 : Matrix3d;
                  BEGIN

CreateYRotationMatrix(CenterX,CenterY,
                      CenterZ,Azimuth,Trans1);

CreateZRotationMatrix(CenterX,CenterY,CenterZ,AOE,Trans2);
                       MatrixMultiply(Trans1,Trans2);
        {(B*C)}
                       M11 := Trans1[1,1];
                       M21 := Trans1[2,1];
                       M31 := Trans1[3,1];
                       M41 := Trans1[4,1];
                       M12 := Trans1[1,2];      {make simple
values out}
```

```
                        M22 := Trans1[2,2];      {of matrix
values for speed}
                        M32 := Trans1[3,2];
                        M42 := Trans1[4,2];
                        M13 := Trans1[1,3];
                        M23 := Trans1[2,3];
                        M33 := Trans1[3,3];
                        M43 := Trans1[4,3];
END;
{this procedure transfers the array to an array of integer.
Scale_By sets the nodal size }

PROCEDURE RoundAndTransfer(PotShadePts : Integer;
                           X,Y,Z       :   SelectPtr;
                           Id          :   IntSelectPtr);
          VAR
             Ct : Integer;
          BEGIN
               IF Scale_By <> 1.0 THEN BEGIN
                  FOR Ct := 1 TO PotShadePts DO BEGIN
                      SelectIntCoord[Ct,1] := Round(X^[Ct] *
                                              Scale_By);
                      SelectIntCoord[Ct,2] := Round(Y^[Ct] *
                                              Scale_By);
                      SelectIntCoord[Ct,3] := Round(Z^[Ct] *
                                              Scale_By);
                      SelectIntCoord[Ct,4] := Id^[Ct];
                  END;
               END
               ELSE BEGIN
                    SelectIntCoord[Ct,1] := Round(X^[Ct]);
                    SelectIntCoord[Ct,2] := Round(Y^[Ct]);
                    SelectIntCoord[Ct,3] := Round(Z^[Ct]);
                    SelectIntCoord[Ct,4] := Id^[Ct];
               END;
END;


{Procedure ShadeSearch efficiently searches a rounded set of
coordinate points to determine the number of shading objects
for each target point}

PROCEDURE ShadeSearch(ObsNo          : Byte;
                      Targets,Points : Integer;
                      SelfShade      : Boolean);
VAR
   TargetCoord             : IntTargetArr;
   TempCt,Ct,Ct2,X,Y,Z,Pos : Integer;
BEGIN
```

```
     {SelectIntCoord and ClusterInfo are all Global Var.}
     {Move clusters into a separate array}

     WordMove(SelectIntCoord,TargetCoord,Succ(Targets) *
SizeOf(IntCoordinates));

     {if targets are to be removed as potential shade
      sources (self shade)}
     {then we remove then from the shade array}

     IF NOT SelfShade THEN BEGIN
           IF Points > Targets THEN BEGIN

WordMove(SelectIntCoord[Succ(Targets)],SelectIntCoord[1],
                  (Points - Targets) *
SizeOf(IntCoordinates));
              Dec(Points,Targets);
           END
           ELSE
               Exit;    {there is no shade!}
      END;

     {Sort both arrays by Y}

     SelectIntSort(Points,SelectIntCoord);
     TargetIntSort(Targets,TargetCoord);

     TempCt := 0;
     FOR Ct := 1 TO Targets DO BEGIN
           Ct2 := TempCt;
           Y := TargetCoord[Ct,2];
            REPEAT
                Inc(Ct2);
            UNTIL (SelectIntCoord[Ct2,2] = Y) OR (Ct2 >
                  Points);
            IF Ct2 <= Points THEN BEGIN
               TempCt := Pred(Ct2);
               Z   := TargetCoord[Ct,3];
               X   := TargetCoord[Ct,1];
               WHILE (Ct2 <= Points) AND
                     (SelectIntCoord[Ct2,2] = Y) DO BEGIN
                     IF (SelectIntCoord[Ct2,3] = Z) AND
                        (SelectIntCoord[Ct2,1] > X) THEN
                        Inc(ClusterInfo[TargetCoord[Ct,4]]);
                     Inc(Ct2);
               END;
           END;
     END;
END;
```

```
{This procedure calls all of the simulation routines}

PROCEDURE GetShadePts(X,Y,Z             :ArrPtr;
                      Id                : IntArrPtr;
                      TotalPts,Targets  : Word;
                      AOE,Azimuth       : Single;
                      CenterX,CenterY,
                      CenterZ           : Single;
                      DefineTarget      : TargetMaxMin;
                      ObsNo             : Byte;
                      SelfShade,NewSim  : Boolean);
          {here we use dsc to hold values from call to call}
          const
               OldAz      : single =  0; {an invalid number}
               AzPotShade : Integer = 0;
          var
               PotShade,ByteNo    : Integer;{an arbitrary #}
               SuccTargets        : Integer;
          BEGIN
               if NewSim then OldAz := -9999;

Movetargets(Targets,X,Y,Z,Id,TempX2,TempY2,TempZ2,TempId2);
               If Azimuth <> OldAz then begin
                 ClipByAz(-Azimuth,Targets,TotalPts,
                          X,Y,Z,Id,TempX,TempY,
                          TempZ,TempId,DefineTarget,
                          AzPotShade);
               END;
               PotShade := AzPotShade;
               if AOESearchSw(-AOE,-Azimuth) then
                 ClipByAOE(-AOE,Targets,PotShade,
                           TempX,TempY,TempZ,TempId,
                           TempX2,TempY2,TempZ2,TempId2,
                           DefineTarget,PotShade)
               else begin
                    SuccTargets := Succ(Targets);
                    ByteNo := (PotShade - Targets) *
                              SizeOf(Single);
                    WordMove(TempX^[SuccTargets],
                             TempX2^[SuccTargets],
                             ByteNo);
                    WordMove(TempY^[SuccTargets],
                             TempY2^[SuccTargets],
                             ByteNo);
                    WordMove(TempZ^[SuccTargets],
                             TempZ2^[SuccTargets],
                             ByteNo);
                    ByteNo := (PotShade - Targets) * 2;
                    WordMove(TempId^[SuccTargets],
                    TempId2^[SuccTargets],
```

```
                    ByteNo);
                END;
                Rotate(CenterX,CenterY,CenterZ,Azimuth,AOE);
                VectorMatrixMultiply(PotShade,TempX2,
                                     TempY2,TempZ2);
                RoundAndTransfer(PotShade,TempX2,TempY2,
                                 TempZ2,TempId2);
                ShadeSearch(ObsNo,Targets,PotShade,
                            SelfShade);
END;
BEGIN
END.

                (*****PROGRAM SIM.PAS*****)

{Sim requires 2 input files: the sun locations to simulate
and afile containing the location of the clusters, the
location of the target and the light penetration per
cluster.
The file that locates the sun must be in the following
format:

Number of positions to simulate (integer)
Angle Azimuth (real)
.
.
.
The file containing the cluster locations must have the
following format:

Total number of nodes (integer)
Number of nodes in the target (integer)
TargetX TargetY TargetZ LightPen (real)
.
.
.
NonTargetX NonTargetY NonTargetightPen (real)
.
.
.
Notice that the target nodes are listed first.

}

{$V-,I-}
USES Global,Crt,ScreenIO,
     Library;
Var
   totalObs,TargetNo,TotalNo : word;
   NodalDist                 : Single;
```

```
{clean things up upon Exit}

{$F+}PROCEDURE MyExit;{$F-}
               BEGIN
                    SetCursor(Cursor);
                    TextAttr := 7;
                    Release(HeapOrg);
                    ExitProc := ExitVect;
END;


{**The simulation starts here**}

PROCEDURE FindCenter(Targetdef                 : TargetMaxMin;
                  VAR CenterX,CenterY,CenterZ,
                      Dimension                : Single);
          BEGIN
               CenterX := TargetDef[MinX] +
                         (TargetDef[MaxX] - TargetDef[MinX])
                          * 0.5;
               Dimension := TargetDef[MaxY] -
                            TargetDef[MinY];
               CenterY := TargetDef[MinY] + Dimension * 0.5;
               CenterZ := TargetDef[MinZ] +
                         (TargetDef[MaxZ] - TargetDef[MinZ])
                          * 0.5;
               IF Scale_By <> 1.0 THEN BEGIN
                  CenterX := CenterX * Scale_By;
                  CenterY := CenterY * Scale_By;
                  CenterZ := CenterZ * Scale_By;
                  Dimension := Dimension * Scale_By * 0.8;
               END;
END;
PROCEDURE Simulate(SelfShade : Boolean);
          VAR
             TargetDefinition : TargetMaxMin;
             PotentialShadePts : Integer;
             Ct                : Integer;
             TAz,Cx,Cy,Cz,Dim  : Single;
             FirstSim          : Boolean;
          BEGIN
               FirstSim := True;
               if FirstSim then begin
                  FindMaxMin(Xd,Yd,Zd,
                             TargetNo,TargetDefinition);
                  FindCenter(TargetDefinition,Cx,Cy,Cz,Dim);
               END;
               FOR Ct := 1 TO TotalObs DO BEGIN
```

```
                    TAz := 270.0 - Az^[Ct];
                   {adjust azimuth to cartesian space}
                    GetShadePts(Xd,Yd,Zd,IdVect,
                                LightPen,TotalNo,TargetNo,
                               -AOE^[Ct],-TAz,
                                ACIntensity^[Ct],
                                0,0,0,TargetDefinition,
                                Ct,SelfShade,FirstSim);
                    FirstSim := False;
                  END;
END;
FUNCTION Exists(VAR FileName : String): Boolean;
          VAR
             InFil : File;
          BEGIN
               Assign(InFil,FileName);
               ReSet(InFil);
               IF IOResult <> 0 THEN
                  Exists := False
               ELSE BEGIN
                    Close(InFil);
                    Exists := TRUE;
               END;
END;
PROCEDURE FileError(Err : Integer;VAR FileType : Text);
          BEGIN
               if Err <> 0 THEN BEGIN
                  WriteLn('Error in File Read');
                  Close(FileType);
                  Halt;
                END;
END;
PROCEDURE NonExistant(VAR FileType : Text);
          VAR
             Ch : char;
          BEGIN
               WriteLn('File does not exist - Repeat Input?
                        (Y/N)');
               Ch := CharInput(['Y','N']);
               IF Ch <> 'Y' THEN BEGIN
                  Close(FileType);
                  Halt;
                END;
END;
PROCEDURE DataInput;
          VAR
             Infil    : Text;
             fileName : String;
             Ct       : Word;
          Label
```

```
            PtFilInp,SunFilInp,Jmp;
        BEGIN
            ClrScr;
        PtFilInp:
            Write('The Point File        : ');
            ReadLn(FileName);
            IF Exists(FileName) THEN BEGIN
               Assign(Infil,FileName);
               ReSet(InFil);
               ReadLn(InFil,TargetNo);
               ReadLn(InFil,NodalDist);
               FileError(IOResult,Infil);
               Ct := 1;
               WHILE NOT EOF(InFil) DO BEGIN
                     IF Ct = Max_Pts THEN BEGIN
                        WriteLn('Incomplete Point Set');
                        GoTo Jmp;
                     END;
                  ReadLn(InFil,Xd^[Ct],
                         Yd^[ct],Zd^[Ct],LightPen^[Ct]);
                     IdVect^[Ct] := Ct;
                     Inc(Ct);
                     FileError(IOResult,Infil);
               END;
               TotalNo := Pred(Ct);
            END ELSE BEGIN
                NonExistant(InFil);
                GoTo PtFilInp;
            END;
          Jmp:
            Close(InFil);
      SunfilInp:
            Write('The Sun Position File : ');
            ReadLn(FileName);
            IF Exists(FileName) THEN BEGIN
               Assign(InFil,fileName);
               Reset(InFil);
               ReadLn(InFil,TotalObs);
               FileError(IOResult,Infil);
               FOR Ct := 1 TO TotalObs DO BEGIN
                   ReadLn(InFil,Az^[Ct],
                          AOE^[ct],ACIntensity^[Ct]);
                   FileError(IOResult,Infil);
               END;
               Close(InFil);
            END ELSE BEGIN
                NonExistant(InFil);
                GoTo SunFilInp;
            END;
END;
```

```
PROCEDURE Output;
          Var
             Ct1,Ct2   : Word;
             OutFil    : Text;
             FileName : String;
            Label
               LOOP;
          BEGIN
          LOOP:
               TextAttr := HiVid;
               Clrscr;
               GoToXY(20,12);
               Write('Enter the filename : ');
               SetCursor(Cursor);
               ReadLn(FileName);
               IF Exists(FileName) THEN BEGIN
                  CenterText(14,Reverse,'File Already
                             Exists, Overwrite ? (Y/N)');
                   IF CharInput(['Y','N']) = 'N' THEN
                      GoTo LOOP;
               END;
               Assign(OutFil,FileName);
               ReWrite(OutFil);
               FileError(IOResult,OutFil);
               WriteLn(OutFil,TotalObs);
               FOR Ct1 := 1 TO TargetNo DO BEGIN
                   FOR Ct2 := 1 TO TotalObs DO BEGIN
                       Write(OutFil,
                             ClusterInfo[Ct1].
                             ShadeArr[Ct2]:10:4);
                       FileError(IOResult,OutFil);
                   END;
                   WriteLn(OutFil);
                END;
                Close(OutFil);
END;
PROCEDURE InitializeClusters(Targets,Observations : Word);
          VAR
             Ct1,Ct2 : Word;
          BEGIN

FillChar(ClusterInfo,Sizeof(ClusterInfo),0);
                FOR Ct1 := 1 to Targets DO
                    FOR Ct2 := 1 to Observations DO
                        ClusterInfo[Ct1].ShadeArr[Ct2] :=
                            ACIntensity^[Ct2];
END;
PROCEDURE GetMenus;
          VAR
             SelfShade : Boolean;
```

```
            Ct,SimNo  : Word;
          BEGIN
               ExitVect := ExitProc;
               ExitProc := @MyExit;
               New(Az);
               New(AOE);
               New(ACIntensity);
               New(Xd);
               New(Yd);
               New(Zd);
               New(LightPen);
               New(IdVect);
               New(TempX);
               New(TempY);
               New(TempZ);
               New(TempId);
               New(TempX2);
               New(TempY2);
               New(TempZ2);
               New(TempId2);
               Write('Include self shade? (Y/N) : ');
               SelfShade := (CharInput(['Y','N']) = 'Y');
               writeLn;
               DataInput;
               InitializeClusters(TargetNo,TotalObs);
               Scale_By := 1/NodalDist;
               Wait;
               Simulate(Selfshade);
               Output;
END;
BEGIN
     GetMenus;
END.
```

BIOGRAPHICAL SKETCH

Kevin Scot McKelvey was born on November 10, 1956. He lived in Colorado and Montana before coming to Florida in 1986. He received a high school diploma in 1974 and a Bachelor of Arts degree in history with minor in English from the University of Montana in 1984. He became interested in forestry while working for the Forest Service from 1978-1986 in the summers. He received a Master of Science degree in forestry from the University of Montana in 1986. The field of specialization was entomology/pathology. He has received a Bright Memorial Fellowship, a Montana Woods Products Association Scholarship, and a W.P Shelley Fellowship.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Katherine C. Ewel, Chair
Professor of Forest
Resources and Conservation

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Loukas G. Arvanitis
Professor of Forest
Resources and Conservation

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

George E. Bowes
Professor of Botany

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Henry L. Gholz
Professor of Forest
Resources and Conservation

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Jimmy W. Jones
Professor of Agricultural
Engineering

This dissertation was submitted to the Graduate Faculty of the School of Forest Resources and Conservation in the College of Agriculture and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

August, 1990

Director, School of Forest Resources and Conservation

Dean, Graduate School

UNIVERSITY OF FLORIDA
3 1262 08556 8128